AFRL-IF-RS-TR-2000-6
In-House Report
March 2000

DATA EMBEDDING FOR COVERT COMMUNICATIONS, DIGITAL WATERMARKING, AND INFORMATION AUGMENTATION

Arnold C. Baldoza, 1Lt, USAF

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2000-6 has been reviewed and is approved for publication.

APPROVED:

GERALD C. NETHERCOTT
Chief, Multi-Sensor Exploitation Branch

FOR THE DIRECTOR: John V McNamara

JOHN V. MCNAMARA, Technical Advisor
Information & Intelligence Exploitation Division

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFEC, 32 Brooks Road, Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 2000

3. REPORT TYPE AND DATES COVERED
In-House, Aug - Nov 99

4. TITLE AND SUBTITLE
DATA EMBEDDING FOR COVERT COMMUNICATIONS, DIGITAL WATERMARKING, AND INFORMATION AUGMENTATION

5. FUNDING NUMBERS
PE - 62702F
PR - 3480
TA - PR
WU- OJ

6. AUTHOR(S)
Arnold C. Baldoza, 1Lt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFEC
32 Brooks Road
Rome, NY 13441-4114

8. PERFORMING ORGANIZATION REPORT NUMBER
AFRL-IF-RS-TR-2000-6

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFEC
32 Brooks Road
Rome, NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2000-6

11. SUPPLEMENTARY NOTES
AFRL Project Engineers: Arnold C. Baldoza, 1Lt USAF, 315-330-7838 and Richard Simard, 315-330-1798, IFEC.
CSE 996 Master's Research Project (Dr. Kamal Jabbour, Graduate Advisor), Syracuse University.

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This paper introduces data embedding schemes for covert communications, information augmentation, and digital watermarking. Although most digital objects may be used as a container for embedded data, this paper focuses on digital images. It discusses the data embedding framework -- the assumptions made on the communication channel, the issues considered for implementation, and the general requirements imposed on data embedding techniques. The paper presents a number of spatial and spectral data embedding techniques and touches on the general limitations these techniques face. Several application scenarios are presented and the manner in which the data embedding framework is typically tailored and applied to them is proposed.

14. SUBJECT TERMS
steganography, data embedding, information hiding, digital watermarking, information warfare, secure communication

15. NUMBER OF PAGES
96

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

Acknowledgements

The author acknowledges the support of the Air Force Research Laboratory, Information Directorate, Information and Intelligence Exploitation Division, Multi-Sensor Exploitation Branch. This paper would not have been possible without the generous support of Mr. Gerald Nethercott and Mr. Richard Simard. The author also acknowledges the support provided by Dr. Jiri Fridrich of the State University of New York in Binghamton. Dr. Fridrich generously provided the Matlab code to [Ruan98] and [Giro89] and provided invaluable technical insight into the world of data embedding.

1 Introduction to Data Embedding

[Ande98] offers two reasons for the rapid growth of interest in data embedding techniques since 1996. First, an appreciation of new market opportunities created by digital distribution is coupled with a fear that digital works are too easy to copy. Traditional art can be authenticated by studying, for example, paint brush strokes. On the other hand, digital works do not lend themselves to similar types of analysis. Because of this, the publishing and broadcasting industries have become interested in techniques for embedding copyright marks and serial numbers in digital films, audio recordings, books, and multimedia products. Second, policy moves by various governments to restrict the availability of encryption services have motivated people to study methods by which private messages can be hidden in seemingly innocuous cover messages. The current basis for export control for the United States – the Wassenaar Arrangement of Export Controls for Conventional Arms and Dual-Use Goods and Technologies (adopted on 13 July 1996) – includes cryptographic products on its export control list. Although this stance has been softened by recent executive and legislative rulings [McCu99, Fran99], it has prompted the use of data embedding tools to conceal information from casual interception.

The field of data embedding has its infancy in steganography. The word steganography comes from the Greek *steganos* (covered or secret) and *-graphia* (writing or drawing) and thus means, literally, covered writing. Steganography serves to hide secret messages in other messages such that the existence of the secret message is concealed. Historical steganographic techniques include invisible ink, tiny pin punctures on selected characters, minute differences between handwritten characters, pencil marks on typewritten characters, and so on [Kahn96]. However, steganographic techniques do not have to be technically sophisticated to be successful. It is common to hide messages in the first letters from words or sentences of some innocent looking text. In addition, one could write a letter or create a painting that appears inconspicuous to an observer. [Fran96] gives the simple following example: one could send write a letter to his brother containing a painting from his three-year-old daughter -- the painting of an apple tree. A normal observer, like a policeman, would see the picture. The covering letter will tell him that the proud father has sent the painting of his daughter. However, the brother would count the number of apples on the picture and know: "Our meeting will be at 3 pm."

It is an erroneous to equate steganography to cryptography. Cryptography deals with the process of disguising messages such that unauthorized persons cannot deduce the substance of the message. Anyone eavesdropping on the communication channel will be able to see the exchange of ciphertext (encrypted message), but without the secret key, will not be able to retrieve the original plaintext (intelligible message). Encryption provides confidentiality, but sometimes this is not enough. For example, the detection of encrypted message traffic between a military person and a hostile government has obvious implications. Steganography is the art and science of communicating through covert channels. Steganography serves to hide (secret) messages in other messages such that the existence of the (secret) message is concealed. In the simplest implementation, the sender writes an innocuous message and then conceals a secret message on the same piece of paper.

The field of embedding data has evolved from the ancient art of steganography. Data embedding (also known as information hiding) informally refers to "hidden writing", i.e., writing that is not readily discernible to the casual observer. This is a more general case of steganography. In steganography, the presence of a message is concealed. This is not a requirement for more general data embedding techniques.

With the growing ubiquitous nature of the Internet and of electronic correspondence, steganographic techniques naturally evolved to use digital images, digital video, or digital audio for hiding messages. The applications for these techniques have widened to include more than covert communications. There are three major areas of data embedding applications. These areas are differentiated not by the techniques that they use, but by the relationship between the embedded data and the object wherein which the data is embedded into. These areas of applications are (1) covert communications (or steganography), (2) information augmentation (or captioning), and (3) document marking (Figure 1). Document marking can be further subdivided into source and content authentication. In covert communications applications, the embedded data has no relationship with the embedded data. Undetectability of the embedded data is paramount. In captioning applications, the embedded data provides additional information about the object into which it was embedded. Undetectability is not a concern, although capacity is typically maximized. In addition, robustness is not an issue since there is no motivation for someone to remove the additional information. In document marking for source authentication, the embedded data provides ownership information. Unlike in a covert communication scenario, undetectability and capacity is not significant. It is understood that every principal with access to the object knows that it has been marked. As such, robustness to attempts to remove the ownership information is critical. It is assumed, especially in copyright protection scenarios, that there exists an unscrupulous person who is motivated to remove embedded ownership information. There is a limit on what this attacker can do to the image, however, since he or she also wishes to preserve the quality of the image. In a content authentication application, tamper detection seals are embedded into the image. These seals "break" whenever the content of the image is modified. The various requirements for these three application areas will be further discussed in this paper.

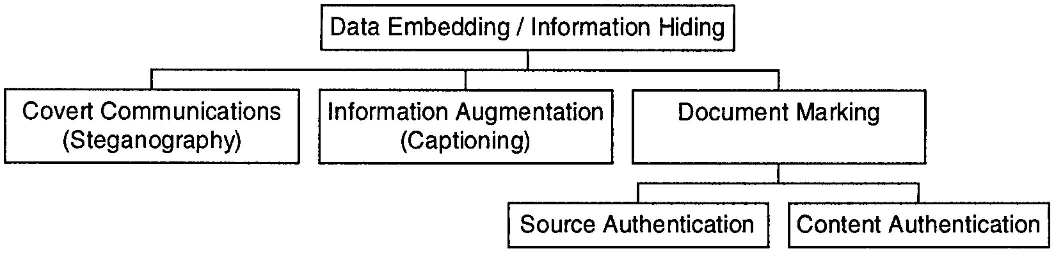

Data Embedding / Information Hiding
Covert Communications (Steganography)
Information Augmentation (Captioning)
Document Marking
Source Authentication
Content Authentication

Figure 1. Application Areas of Data Embedding Techniques.

In all applications, data embedding involves imperceptibly embedding marks or data into a digital media object to enhance or protect its value. The embedded data may contain information such as authentication or copyright codes. Although the marked media is perceptually unchanged, the embedding method imposes detectable or extractable modifications, given the appropriate extraction algorithm. Ideally, the imposed modification does not degrade the host beyond its utility value.

Although most digital objects in the digital universe can be processed to hide additional information, this paper deals with a subset of these digital objects. More specifically, this paper deals with information hiding techniques in digital images.

Data embedding terminology was standardized in the First Information Hiding Workshop, which was held in the United Kingdom in 1994. Whenever possible, this paper will use the same terminology. The data that is embedded into the digital object is referred to as the *embedded data* or as the *plain object*. The digital object into it is embedded into is considered to be the *cover object*. Embedding the plain object into the cover object results into the *stego or marked object* (Figure 2). A *stego-key* or *marking key* is normally used to control the embedding process and to restrict the recovery of the embedded data to principals who know it. Although the terms *stego object* and *stego-key* are reminiscent of pure steganographic techniques, in this paper, they are used in the general data embedding framework.

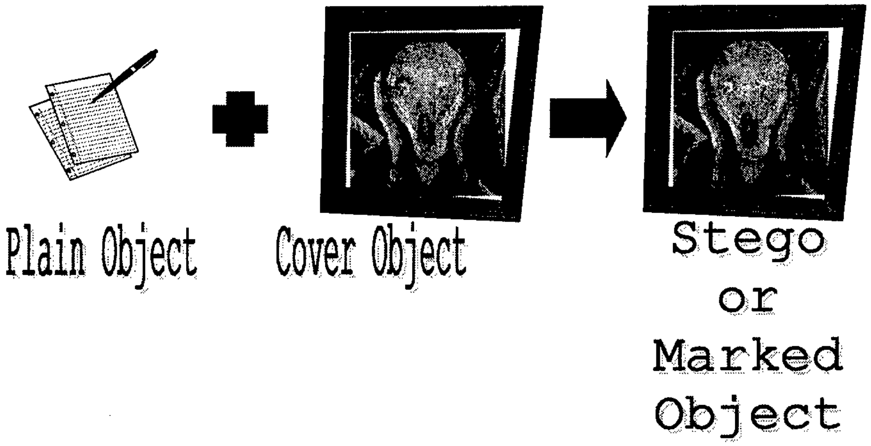


Figure 2. Data Embedding Terminology.

2 Data Embedding Framework

There is no comprehensive theory of steganography in the same manner Shannon treated the theory of encryption [Shan48] or Simmons considered authentication systems [Simm83]. Recently, however, work in information theoretic is being conducted for the data embedding communication channel [Cach98, Mitt99]. Informally, all elements that comprise the data embedding framework can be illustrated using the "Prisoner's Problem" formulated by Simmons [Simm83] in 1983[1]. In this scenario, two principals -- Alice and Bob -- are in prison and they wish to hatch an escape plan. All of their messages pass through Willie, the warden. If Willie detects any encrypted or suspicious message, he will frustrate their plan. Therefore, Alice and Bob must find some way of hiding their escape plans (plain objects) in an innocuous-looking exchange of public messages (cover objects). Suppose the two prisoners communicate by sending digital images. If Alice and Bob can hide their (private) messages in the digital images that they exchange in such a way that Willie is not aware of the private communication, they can successfully plan their escape. If Willie manages to detect the presence of a covert message (through statistical analysis, for example), Alice and Bob are severely penalized. Figure 3 illustrates the elements of this communication framework.

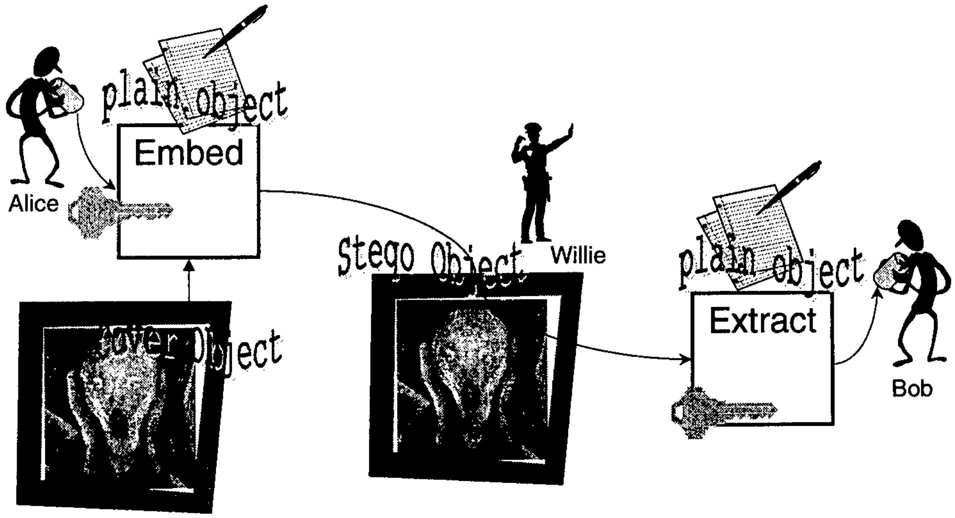


Figure 3. General Data Embedding Framework.

[Mitt99] proposes an information-theoretic approach to data embedding (Figure 4). More formally, a data embedding system is used to transmit (secret) message **V** from a sender (Alice) to a destination (Bob) in such a way that an intermediate party (Willie) is not able to notice that

[1] http://www.conceptlabs.co.uk/alicebob.html has a humorous, yet accurate, treatment of the Prisoner's Problem.

the stego object **X** contains the hidden message[2]. If Willie has control over the communication channel, he can modify a suspect message **X** and transform it into a modified version **Y**. If such modification attacks might occur, the data embedding system should be robust against small distortions in the sense that the embedded information will still reach Bob.

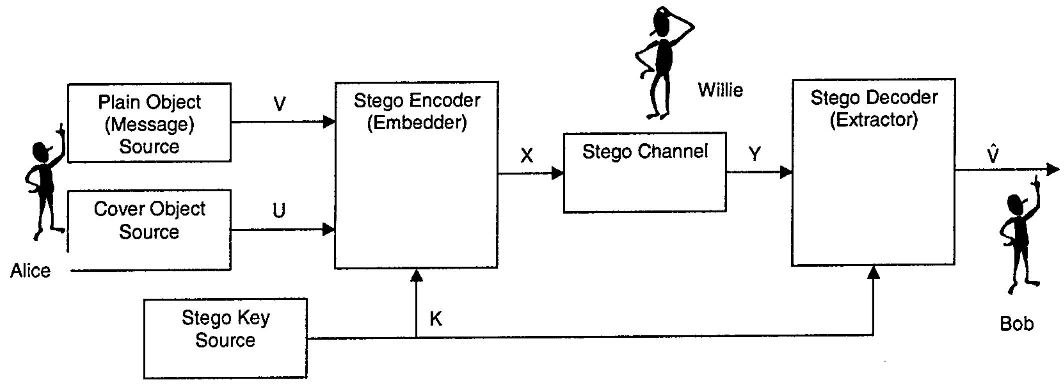


Figure 4. Formal Model for a Data Embedding Framework [Mitt99].

In the data embedding scheme, a (secret) message **V** is embedded in some cover object. The embedding of the message is performed by the data embedding encoder which, depending on some secret key **K**, merges the message **V** into the cover object **U**. For each key value **k**, the stego encoder $f_k(.,.)$ produces the stego object $X = f_k(U,V)$. It is assumed that the encoder has an encoder inverse $g_k(.,.)$, i.e., $g_k(f_k(U,V),U) = V$. The stego object should look genuine, i.e., the stego object should not be distinguishable from a typical message of the message source. A possible way of imposing this condition mathematically is by choosing a suitable distortion measure $d(.,.)$ and by requiring for every key value **k**, the *encoder constraint*

$$Ed(U,X) \leq D'$$

[2] It is important to note that Willie does not "notice" that the stego object contains a hidden message. This imposes that the perceptual quality of the image is maintained. Undetectability through statistical methods may or may not be a requirement.

where the expectation operation **E** is with respect to the joint probability distribution on **U** and **X**. The bound is chosen suitably small to guarantee that the stego object **X** is essentially indistinguishable from the cover data **U**.

In a covert communication application, the steganographic encoder should not introduce any statistically detectable artifacts when transforming **U** into **X**. In digital watermarking applications, the embedding of the secret message should be robust in addition to satisfying the encoder constraint. This robustness requirement can be modeled by a data embedding channel in the following way. The attacker is allowed to modify the stego object only in a limited way, otherwise the quality of **X** will suffer too much -- a distorted image **Y** after a distortion attack must still have reasonable quality. This quality requirement, which will be called the *channel constraint*, can be expressed as

$$Ed(X,Y) \leq D''$$

where the expectation is with respect to the joint probability distribution on **X** and **Y**. This implies that the distorted image **Y** must be close to the stego image **X** for small D''. The channel constraint can also be generalized to include geometric transformations by the attacker.

It is implicitly assumed that the attacker has no knowledge of the secret message **V**, the secret key **K**, or the source message **U** other than that contained in her observation **X**. For instance, if he knew **U**, he could set **Y=U** and deceive the receiver.

The goal of the decoder is to reconstruct the secret message **V** from the received (distorted) message **Y** using the secret key **K**. In many cases, it is assumed that the decoder has no knowledge of the cover message **U** (oblivious scheme). However, there are applications where the decoder has access to the cover message (non-oblivious scheme). Knowledge of the cover message can help in the decoding process. One can utilize the cover image to pre-process the (distorted) stego object before invoking the decoding processes. In any case, an ideal decoder should not give an attacker information on how to remove the embedded data. [Kalk, Kalk97] has shown that a watermark detector -- if not properly designed -- can be used to remove a digital watermark.

As in cryptography, we assume that the security should depend on a secret key that Alice and Bob have somehow managed to share. The algorithm for the data embedding encoder is assumed to be public knowledge, even though in practice this would be rather difficult, if not almost impossible, to obtain this knowledge[3]. In a watermarking application, security also rests on the inability of an attacker to remove the embedded watermark or render it unreadable without degrading the quality of the stego object. In a covert communication application, any artifact introduced into the cover object by the steganographic encoder must be statistically undetectable.

2.1 Data Embedding in Digital Images

This paper deals with only a subset of digital objects that can be used as a container for embedded data. More specifically, this paper deals with information hiding techniques in digital images. In its raster data format, digital images are represented by a two-dimensional array (e.g., a matrix) in which each element of the matrix corresponds to a single pixel[4] in the displayed image. Each pixel may or may not be physically square. The element value in the matrix is usually determined by the type of the image. There are four basic types of images: (1) binary images, (2) intensity images, (3) indexed images, and (4) RGB images. The differences of these different types are summarized in Table 1.

[3] This corresponds to the Kerkhoffs principle of cryptography [Kerk1883].

[4] The term "pixel" is derived from "picture element." Physically, it corresponds to a single dot on a computer display.

Table 1. Different Types of Digital Images.

Type of Image	Description
Binary Images	Each pixel assumes one of only two discrete values: on (white) and off (black). Binary images are typically unsuitable to be a cover image because of their extremely low color depth and their intolerance to noise adding.
Intensity Images	Each pixel is represented by an intensity value between 0 (black) and 255 (white). Intensity images offer moderate capacity for embedded data. For example, if a least significant bit encoding scheme was used, one can embed $(M \times N/8)$ 8-bit data symbols into an N×M intensity image. Increase a pixel value by one, in a least significant bit encoding scheme, introduces a 0.39% change in its intensity. This change is visually imperceptible.
RGB Images	Similar to intensity images. A pixel can be represented in two different ways: (1) as a tuple of three intensities (red, green, and blue), or (2) as three separate color planes. RGB images offers as much as three times the capacity of similar intensity images.
Indexed Images	Consists of two arrays, an image matrix and a colormap (or palette). The colormap is an ordered set of values that represents the colors in the image. For each image pixel, the image matrix contains a value that is an index into the colormap. The most common of this type of image are CompuServe's Graphics Interchange Format (GIF) images. Data embedding techniques for indexed images embed data either in the image's palette or in the image's data.

Data embedding in digital images is possible because the gaps in human visual systems can be easily exploited. In the case of images, the human eye is relatively insensitive to high frequencies. This fact has been utilized in many data embedding algorithms. For example, one can modify the least significant bit of the gray levels in a digital image to embed a message. Least significant bit encoding is a high frequency operation that introduces as little as a $1/256$ change to roughly 50% of an image's pixels. The human eye is also relatively insensitive to gradual changes in shades. Because of this, bits of information can be embedded into coefficients of image transforms, such as the Discrete Cosine Transform or the Fourier Transform, which represent can signals as a weighted summation of sine and cosine functions.

Studies also show that the human eye is less sensitive to changes in the blue channel in an RGB image and to the overall luminance channel [Hons98]. These properties of the human visual system can be exploited to hide information into a digital image. It must be pointed out, however, that exploiting the gaps of the human visual systems can only be successful if the ultimate consumer of the image is the human eye. Imperceptibility, which is related to the human detector, is not equivalent to undetectability, which belongs to the field of statistics. Simply because a human cannot visually perceive an embedded message does not necessarily imply that a computer cannot find a statistical anomaly introduced by the embedded information in the image.

Although the general data embedding scenario is simple, assumptions regarding the communication channel, implementation issues, and data embedding requirements must be formalized and addressed before any discussion about data embedding techniques can be meaningful. As one will see, the relationship between the assumptions made, the requirements imposed, and the design implemented interact with one another. Therefore, tradeoffs are always weighed.

2.2 Assumptions on the Communication Channel

One assumption is the stability of the communication channel. Is Willie a passive or active warden? A passive warden merely attempts to discover the presence (perhaps in a probabilistic setting) of embedded data in the cover object. An active warden tries to reduce the channel capacity of the covert channels in order to deny any hidden message from reaching its intended recipient. The assumption made on the stability of the communication channel directly affects the robustness requirement (a.k.a. data persistence) on the data embedding technique. If a passive warden is assumed, robustness is not an issue -- it is assumed that the stego object will not be subject to distortion or removal attacks. If an active warden is assumed, it is assumed that the stego object will undergo some types of distortion and that the plain object must still be recoverable by the recipient.

Another is the assumption of the capacity of the communication channel. How large are the private messages with respect to the images into which these message are embedded? In determining the length of the message, one must differentiate between the information bits (the logical bits that comprise the message) and the data bits (the physical bits that encode the message). A short message (small number of information bits) may be encoded redundantly throughout the image or may be encoded using an error correction code (large number of data bits). The higher the number of physical data bits that comprise the message, the greater chances the message introduces detectable artifacts into the cover object. As such, capacity affects imperceptibility and detectability requirements. Imperceptibility deals with visual artifacts that may be introduced into the cover object. Detectability deals with statistical analysis of the stego object in an attempt to deduce whether or not the object contains an embedded message.

A third assumption is the authentication of the communicating parties. What is the possibility that the (active) warden will attempt to spoof one of the communicators? This assumption lays down the requirements for security and invertibility. Security is normally based on resistance to attacks to obtain the secret key, as it is in a cryptographic framework. However, it also includes attacks that use the knowledge of the embedding and extracting algorithm to obtain, detect, disrupt, or remove the embedded message.

2.3 Data Embedding Implementation Issues

Data embedding schemes can be categorized on how they address certain data embedding implementation issues. These issues are:

1. whether the embedded data is privately or publicly extractable,
2. whether the original image is required to extract the embedded data, and
3. whether the data can be extracted asymmetrically or not.

Private or Public Marking

Private marking systems are characterized in the following manner: one can only decide whether certain data is embedded or not. Private marking schemes, popular among watermarking schemes, typically utilize a simple hypothesis test:

hypothesis H_0: the watermark w is not present
hypothesis H_1: the watermark w is not present

The problem of hypothesis testing is to decide which of the hypothesis is true, when a stego object Y is given. Usually it is not possible to separate all watermarked and unwatermarked objects perfectly: a received object Y might be watermarked with probability $p(H_1|Y)$ or not watermarked with probability $p(H_0|Y)$. We trade off the probability p_{FP} of accepting H_1 when H_0 is true (false positive) and the probability p_{FN} of accepting H_0 when it is false (false negative). Bayes' solution is the decision rule

$$\frac{p_Y(Y \mid H_1)}{p_Y(Y \mid H_0)} \begin{cases} > K \Rightarrow accept \quad H_1 \\ \leq K \Rightarrow accept \quad H_0 \end{cases}$$

where $K = \mathrm{cost}_{pFP} pH_0/(\mathrm{cost}_{pFN} pH_1)$ is a constant depending on the *a priori* probabilities for H_1 and H_0 and the cost connected with the different decision errors [Egge99].

In the copyright protection architecture, for example, given an ownership mark m and an image I, one can either detect or not detect the presence of m in I. Note that with a private marking scheme, there is no means to detect the ownership code if some hypothesis about the possible owner cannot be done. It is only possible to verify if the image belongs to a particular author whose ownership code is known. An example of a private marking scheme is presented in [Cox95a]. A watermark is created using a private key and is embedded into an image. The detector recreates the watermark using the same private key and attempts to detect it within a possibly attacked version of the image.

A weakness of the detectable watermarking scheme is presented in [Adel99]. The result of an ownership dispute is the decision made by an arbitrator after comparing several claims of ownership. This result does not determine the rightful ownership in general, since the rightful copyright holder might not be participating in the corresponding dispute. Take two principals, Alice and Bob, who both claim image *I* belongs to them. In legal arbitrary, Alice and Bob will present their watermarks, w_A and w_B, and Judge Judy will determine which watermark can be detected in *I*. If the image, however, really belongs to a third principal Mark, Judy will not be able to determine that Mark's watermark is in the image.

Techniques that allow the embedded data to be read directly from the image is referred to as being a public marking scheme. Without prior knowledge of the embedded mark or the original image, public marking systems extract *n* bits of information from the marked image. In a copyright protection architecture, the ownership information can be read by anyone without prior knowledge of whose watermark to look for. Information is embedded and subsequently retrieved with a (publicly available) key.

Public marking schemes are characterized by embedding multiple marks that correspond to symbols in an alphabet. These marks may correspond to digital pointers, URLs, or ownership codes. [Piva97] and [Ruan98] present a public marking technique.

Watermark Blindness

Particular attention must be paid attention to the algorithm used to recover the embedded data from the image. In some cases, the embedded data is extracted by comparing the marked version of the object to the non-marked one [Cox95a, Swan96b, Wolf96]. With the original object, one can opt to preprocess the object under test to compensate for any distortion. Although these techniques may be very robust with respect to geometric distortions, the availability of the original multimedia object may not be possible.

Techniques that extract the embedded data without resorting to the comparison between marked and non-marked objects are said to be oblivious or blind. Several blind image watermarking algorithms have been proposed and implemented, but none of them is able to survive geometric manipulations. Current research efforts have suggested embedding a reference mark in addition to the watermark itself [Bend96, Hons98, Pere99]. The reference mark is to be used for identifying the type of geometric distortion the object undergone. The resistance to geometric manipulations (e.g., translation, resizing, rotation, cropping, and warping) is still an open issue and a solution must be found before watermarking techniques are successfully applied to image copyright protection. [John99b] proposed a method for the recovery of original size and appearance of images based on the concept of identification marks. This method does not require the original image, but only a small number of salient image points. With these salient image points, it is possible to undo the affine transformation an image had undergone. However, it is contested that the salient image points required are equivalent to having the original image and therefore, any watermarking scheme based on this method is not truly oblivious. Self-synchronizing digital watermarks are also being explored with hopes of being resistant to geometric distortions without requiring the original image or a database of salient image points [Algh99a, 99b, 99c].

Symmetric or Asymmetric Extraction

A technique is said to be symmetric if the stego-key used in the embedding process is the stego-key used for extraction. Asymmetric techniques use different keys. It is widely believed that asymmetric mechanisms are likely to be significantly more robust than symmetric ones. Once the embedded data has been read with the symmetric key, it will be much easier for an attacker to remove it or to make it unreadable. For example, one can encode the inverse of the embedded data.

[Crav99] contends that the keys used in all watermarking schemes eventually become public knowledge. Therefore, a pure asymmetric extraction technique does not exist. During an ownership dispute, the two parties involved in the arbitration must produce their extraction key in order to determine whose watermark is contained in the object. Their keys now become public and they may be used to remove ownership marks in other previously marked objects. To avoid this eventual publication of secret keys, zero-knowledge detection schemes are currently being investigated to provide a totally private scheme [Crav99].

2.4 Requirements on Data Embedding Schemes

As alluded to in the discussion on the assumptions made on the communication channel, requirements on data embedding techniques are application-dependent and, for every application, these requirements vary in degree and have to be balanced against other competing requirements. As listed in [Piva98], these requirements are:

1. security,
2. imperceptibility,
3. detectability with low error probability,
4. payload,
5. robustness, and
6. invertibility.

In evaluating data embedding schemes, one must consider the requirement tradeoffs that the developers may have made. For example, the ability to discern images with embedded messages is directly influenced by the physical length of the message, as well as the format and content and size of the carrier image. Obviously, the longer the message (high capacity), the larger the modification to the carrier image and the higher the probability that the modifications can be statistically detected (low imperceptibility). Because of this tradeoff, one-bit watermarking schemes have a distinct advantage over an *n*-bit watermarking scheme. The choice of the carrier image is also crucial. Natural photographs with 24 bits per pixel provide the best environment for message hiding, especially if they contain highly textured areas. The redundancy of the data in these areas helps to conceal the presence of the embedded messages. Lossy compressed images, such as JPEG files, are more sensitive to small perturbations of the image data, and pose a challenge for creating a secure data embedding technique with reasonable capacity. Indexed images, although abundant over the Internet, also provide a hostile environment for the steganographer. The limited number of available colors imposed by the finite palette makes the process of message hiding a difficult challenge.

Imperceptibility and robustness also compete with each other. The performance of the watermark detector, for example, generally increases with the energy of the watermarks inserted: the stronger the watermark, the better the watermark detector performs. However, the stronger the watermark, the more likely it is to produce visible artifacts[5]. In addition, if the watermarking scheme is overly conservative to guarantee transparency over a variety of input images, it is likely that the watermarks will not be detected after some distortion operation or attack is applied to it.

Security

The security of a data embedding scheme cannot be based on algorithm ignorance. As for cryptography [Kerk1883], it is well known that the effectiveness of an algorithm can not be based on the assumption that possible attackers do not know how the data has been embedded into the multimedia object. Nevertheless, the robustness of most commercial digital watermarking products is based on such an assumption. Though some of them claim to be exceptionally resistant, by knowing how the watermark encoder and decoder work, it is very easy for a skilled hacker to make the watermark unreadable or undetectable [Peti98].

In covert communication applications, undetectability of the embedded message also falls under the security aspect. Even with the knowledge of the encoder and decoder, one should not be able to detect the artifacts introduced during message embedding. We do not require that the hidden message is recovered for a steganalytic attack to be successful – the hidden message must only be detected for the warden to defeat the steganographer[6].

[5] Perceptual masking has made it possible to increase watermarking strength without increasing the visibility of artifacts, but the problem remains.

[6] The question of then decrypting the recovered message is a classical cryptanalytic question and lies outside the interests of steganalysis proper.

[Fran99] discusses possible attacks in order to evaluate the security of a data embedding system. The paper lists seven types of attacks, categorized into passive and active attacks. Covert communication applications mainly consider passive attacks, as pointed out in [Ande98]. The discussion of active attacks is common for watermarking systems [Petit98].

Passive Attacks

A passive attacker (passive warden scenario) is restricted to the statistical analysis of the stego object in order to detect the presence of an embedded message. The nature and amount of data the attacker has access to define the type of attack (Table 2). [John98, 98b, 98c,West99] provides security analysis and reveals the security flaws of available data embedding software.

Table 2. Description of Passive Attacks on Data embedding Systems.

Type of Passive Attack	Description
Stego-Only Attack	Attacker analyzes the intercepted stego object **X**
Stego* Attack	The user has repeated embedded in the same cover object; attacker has access to all resulting stego objects $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$
Cover-Stego Attack	In addition to the intercepted stego **X**, the attacker has access to the cover object **U**
Emb-Stego Attack	The attacker knows both the plain object (message) **V** and the resulting stego object **X**
Cover-Emb-Stego Attack	The attacker knows the cover object **U**, the plain object **V**, and the resulting stego object **X**

Active Attacks

An active attacker (active warden scenario) is allowed to modify the cover or stego object in order to disrupt the data embedding communication channel (Table 3). Disruption of the channel is achieved if the attacker renders the embedded message unextractable.

Table 3. Description of Active Attacks on Data embedding Systems.

Type of Active Attack	Description
Manipulation of Stego Object	The stego object can be manipulated to prevent the transmission of the embedded message. On one hand, this attack attempts to foil the secret communication. On the other hand, the attacker can analyze the reaction of the attacked parties: if they try to send another possible stego object, it could be a sign that data embedding is being used.
Manipulation of Cover Object	The attacker can try to make his attack easier by introducing potential cover objects into the user's database of cover objects. This attack is similar to the Cover-Stego attack, but the attacker takes an active role in acquiring the cover object.

Imperceptibility[7]

Information embedded into digital imagery must not introduce visible artifacts in the image. This is extremely important for professional photographers and movie producers, especially if the data embedding technique is to be used in a copyright protection architecture. However, the attempt to render, for example, digital watermarks invisible to the human observer conflicts with other requirements. In current watermarking techniques, for example, the strength of the watermark is directly related to its visibility. The greater the strength of the watermark, the greater its visibility and the better the watermark detector performs. The use of perceptual masks has allowed stronger watermarks to be embedded into an image without substantially degrading the image. In a covert communication scenario, imperceptibility (and undetectability) would be paramount even at the expense of capacity.

Most evaluations of proposed data embedding schemes are inadequate in gauging the perceptibility of the produced visual artifacts. Typically they rely on a person (or a group of people) voting on whether or not a deterioration in image quality can be visually perceived. What is desired in this area is an adequate computer model of the human visual system that can be integrated into a data embedding scheme such that the creation of visual artifacts can be minimized or avoided without human interaction. Two human visual models are available in the research arena: one proposed by [Giro89]; one used by [Podi98].

Low Error Probability

Even in the absence of attacks or signal distortions, the probability of failing to detect the embedded data (false negative) and of extracting data when it does not exist (false positive) must be very small. It is unacceptable in a legally binding arbitration for a watermarking scheme, for example, to have a high probability of error. In a watermarking scenario, an acceptable probability of false positives is 10^{-6}.

Payload

When speaking of how many bits are embedded in an image, one must draw a distinction between data bits and information bits[8]. For example, it is sufficient to embed one bit of information for digital watermarking schemes. In a private watermarking scheme, this one bit of information could be the answer to "Does Mr. X own this digital image?" However, most digital watermarking schemes encode this one-bit of information redundantly throughout the image to combat the effects of possible image processing operations on the image. Therefore, the actual amount of data embedded is greater than one bit.

With this in mind, special care must be taken in comparing data embedding techniques. One cannot compare the robustness aspect of a data embedding technique that merely hides one bit of data against another data embedding technique that hides one bit of information (possible encoded *n* times in the image data). [Frid99d] proposes a scheme to compare detectable, one-bit watermarking techniques with readable, *n*-bit watermarking techniques.

[7] Imperceptibility, which relates to the human visual system, is different from undetectability, which relates to the statistical analysis.

[8] The terms "information" and "data" are not interchangeable in this discussion.

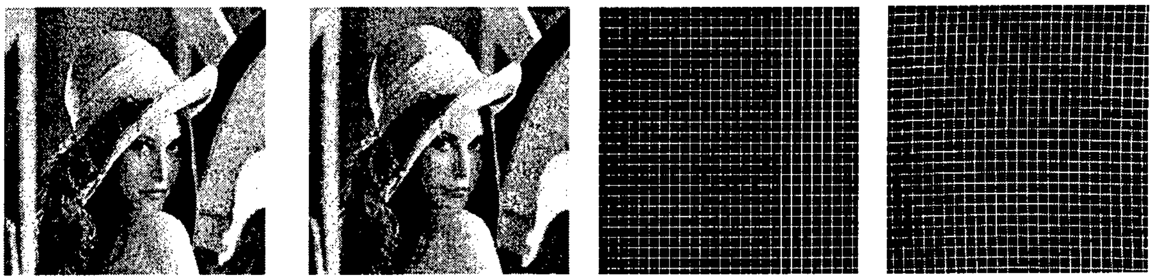
In spite of the importance of knowing how many bits can be embedded into a digital image of a given size, such an issue has not been satisfactory addressed yet. In fact, an upper bound to the amount of information that can be reliably embedded into an image of a given dimension is not known. A number of papers attempt to address this issue [Smit96, Serv98].

Robustness

Digital multimedia objects may undergo numerous types of distortions during their lifecycle. For digital imagery, these distortions include lossy compression, filtering, resizing, contrast and brightness enhancement, cropping, and rotation. Depending on the assumptions made concerning the type of warden in control of the communication channel, embedded data may be required to be persistent even after its cover object had undergone distortions. In the case of digital watermarking for copyright protection, the ownership mark should be detectable even after these distortions occurred.

Resistance to geometric manipulations is very important because they do not severely degrade the quality of the image. They can be applied to make the watermark unreadable. For example, if an image is cropped by one column or by one row or is rotated 0.5 degrees, or scaled to 101%, the difference to the original image is irrelevant and imperceptible. However, the embedded data may no longer be detected or read. Recent efforts have attempted to embed a regular pattern into the image in order to assist in the detection of these geometric distortions [Bend96, aHons98, Adel99]. As mentioned earlier, [John99b] proposed a method for the recovery of original size and appearance of images based on the concept of identification marks. However, it is contested that the availability of the salient image points is equivalent to having the original image. Other efforts have been directed in creating self-synchronizing watermarks [Algh99a, 99b, 99c]. Resistance to geometric manipulations (e.g., translation, resizing, rotation, cropping, and warping) is still an open issue and a solution must be found before watermarking techniques are successfully applied to image copyright protection [Ruan97].

One might attempt to increase the robustness of a data embedding system by trying to foresee all possible attacks. However, [Peti99] points out that given a data embedding system, one can invent a distortion (or a combination of distortions) that will prevent the recovery of the embedded data while leaving the perceptual value of the stego object undiminished. To prove this point and for research purposes, Fabien Petitcolas developed the Stirmark attack which applies a variety of distortions to an image without degrading its quality [Peti98b]. Examples of what Stirmark does are given in Figure 5.



Original Image A　After Stirmark　Original Image B　After Stirmark

Figure 5. Results of a Stirmark Attack.

[Peti99] offers a good explanation of Stirmark:

> Stirmark applies minor unnoticeable geometric distortions. The image is slightly stretched, sheared, shifted, bend, and rotated by unnoticeable random amounts. A slight random low frequency deviation, which is greatest at the center of the image, is applied to each pixel. A higher frequency displacement of the form $\lambda \sin(w_x x)\sin(w_y y)+n(x,y)$ -- where $n(x,y)$ is a random number -- is also added.
>
> Finally, a transfer function that introduces a small and smoothly distributed error into all sample values is applied. This emulates the small non-linear analog-to-digital converter imperfections typically found in scanners and display devices.

For more examples how Stirmark distorts an image, refer to http://www.cl.cam.ac.uk/users/fapp2/steganography/image_watermarking/stirmark/samples2.html.

Invertibility

The definition for watermark invertibility is given by [Crav97]. In their paper, Craver *et al* analyzes the possibility of invalidating ownership claims supported by watermarking by reverse engineering the watermarking process. We say that a watermark is invertible if:

1. It is possible to generate a false watermark and a fake original document which is perceptually equal to the true one, and
2. The false watermark and the true watermark can be found in both the fake original and in the true original.

Take, for example, the following an additive watermarking scheme. Given an image *I*, Alice embeds her watermark, w_A, by

$$I' = I + w_A$$

Then Bob can create a fake original document *I"* by constructing a false watermark, w_B, and subtracting it *I'*.

$$I'' = I' - w_B$$

I" is Bob's forged original image. *I'* contains both watermarks, w_A and w_B, depending on which original image is used to claim ownership.

$$I' = I + w_A = I'' + w_B$$

In addition, Alice's original image contains Bob's watermark and Bob's forged original contains Alice's watermark.

$$I = I'' - w_A + w_B$$
$$I'' = I + w_A - w_B$$

In a legal arbitration, a deadlock ensues since it is not possible to determine which is the real original image and which is the fake one.

For digital watermarking for copyright protection and for source authentication applications, the watermark should not be invertible. Current watermarking techniques use watermarks that are dependent on the image (by using the hash of the image, for example) and on a secret key. This precludes the creation of a false watermark that can be found in the true original by making such creation computationally infeasible.

3 Sample Data Embedding Techniques

This section provides examples of spatial and spectral data embedding techniques. The spatial techniques were mainly developed for covert communication applications where capacity was maximized and a passive warden scenario was assumed. The two spectral examples were developed for digital watermarking applications wherein the embedded watermarks were required to be robust with respect to active attacks. Note that data embedding techniques for one type of image may not be applicable to another type of image. On one hand, data embedding techniques for indexed images cannot be used for intensity images. On the other hand, techniques for intensity images may be modified to work on RGB images. Binary images are typically unusable as cover objects due to their extremely low color depth and their intolerance to noise adding. They still may be used, however, if an embedding technique is developed to use slight geometric distortions to encode a message.

3.1 Spatial Techniques

Data can be embedded in both the spatial domain and the frequency domain. Spatial domain techniques are easy to implement. However, it often fails under signal processing attacks such as filtering and compression. Spatial techniques generally have higher payloads than spectral techniques, but are less robust with respect to most of the signal processing attacks. Additionally, frequency domain techniques tend to have better performance than spatial domain techniques since most perceptual models are developed in the frequency domain [Podi98].

LSB Encoding in Intensity Images

Natural photographs with 24 bits per pixel provide the best environment for message embedding. The redundancy of the data helps conceal the presence of the data [Frid99e]. A simple method of a spatial data embedding technique for 8- or 24-bit intensity images uses least significant bit (LSB) encoding.

Using intensity images as carriers, one can replace N least significant bitplanes of an image with N bitplanes can contain a message[9]. For example, one can replace the least significant bits (N = 1) of a grayscale intensity image with the bits of a message. At most, this would change a pixel's intensity level by 1/256 or 0.39%. With $N = 3$, one replaces the least three significant bitplanes with the bits of a message. This would change a pixel's intensity level by at most 3/256 (1.17%).

Figure 6 illustrates how the replacement of N least significant bitplanes of a grayscale intensity image affects the quality of the image. In each case, N least significant bitplanes were replaced with a random bit pattern of zeros and ones. Even at $N = 3$, which introduces at most 1.17% change in pixel intensity, one can see visual artifacts (especially in areas of smooth, gradual changes) in the resulting images. As in any data embedding technique, one must perform a trade-off between capacity and invisibility. One can embed three times as many bits with $N = 3$ than if N was restricted to $N = 1$.

One can extend this simple LSB encoding by embedding message bits into groups of pixels. For example, one can use odd-even parity to embed the message bit into a group of pixels. Let the parity of a group of pixels be defined as the sum of the individual pixel's parity modulo 2. If the message bit is 1, one forces the parity of the group of pixels to be odd, for example. If the message bit is 0, then one coerces the parity of the group to be even. Because one has the freedom to choose which pixel to modify in order to change the group parity, one can select the pixel that will introduce the least amount of distortion.

[9] Without the loss of generality, a "message" is considered a simple bitstream of data. This bitstream could be a (encrypted) message, another image, or a binary file.

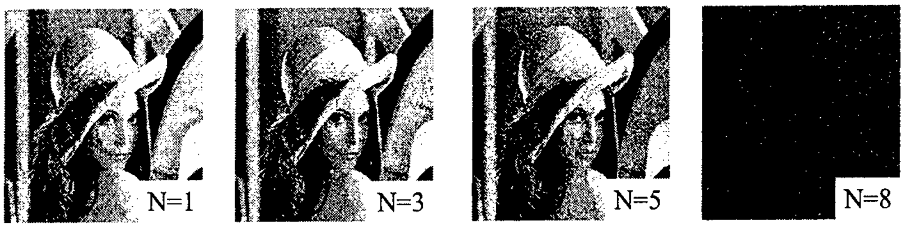


Figure 6. Effects of LSB Encoding.

Data embedding techniques that use LSB encoding are easy to detect [Aura96] and trivial to remove [Ande98]. Almost any trivial filtering operation will affect the value of many of the least significant bits, effectively destroying any message encoded in the least significant bits. To enhance LSB encoding robustness properties, one possible technique is to use redundancy -- either to apply an error correction code or simple embed the message a large number of times. However, an easy way to attack LSB encoding based on this countermeasure is to break up the synchronization needed to locate the samples in which the message is hidden [Ande98]. Digital images, for example, can be cropped, scaled, or rotated.

In some applications, however, the LSB encoding is extreme sensitivity to modification may be a blessing rather than a curse.

Steve Walton proposed an image authentication algorithm based on the fragility of messages embedded in digital images using LSB encoding. In [Walt95], he proposes a technique that uses a key-dependent pseudo-random walk on the image. The checksum is obtained by summing the numbers determined by the seven most significant bits of each pixel along the random walk and taking remainder operation with a large integer *N*. The checksum is inserted in a binary form in the LSB of selected pixels. The checksum is constructed from the seven most significant bits because the LSBs of the pixels used to embed the checksum cannot be allowed to contribute to the checksum calculation.

To check the authenticity of an image, one essentially reverses the embedding process: one extracts the embedded checksum from the selected pixels and calculates the checksum based on the seven most significant bits along the re-generated random walk. If the recovered checksum matches the calculated checksum, the image has not been tampered with.

[Walt95] suggests that this could be repeated for many disjoint random walks or for one random walk that goes through all pixels. To prevent tampering based on exchanging groups of pixels with the same checksum, the checksum can be made “walk-dependent”. The method is very fast and on average modifies only half of the pixels by one gray level. [Bald99] extended this algorithm by subdividing the entire image into sub-blocks and ultimately protecting each sub-block by introducing the walk-dependent checksum. [Bald99] pointed out, however, that this extension could be defeated by simply replacing a block with its corresponding block from another image.

Although checksums can provide a very high probability of tamper detection, they cannot distinguish between a malicious attack (e.g., feature replacement or modification) and an innocent manipulation of the image (e.g., lossy compression, and brightness or contrast adjustment). Increasing the gray scales of all pixels by one would indicate a large extent of tampering, although the image content has been unchanged for all practical purposes.

Data embedding Techniques for Indexed Images

Messages can be embedded into indexed images in either the image palette or the image data. Gifshuffle [Kwan] embeds messages into the palette by permuting its entries. This method does not change the appearance of the image, which is certainly an advantage, but its security is weak because palettes are normally ordered by frequency of occurrence or luminance or some other scalar factor. A randomly ordered palette is suspicious and easily detectable.

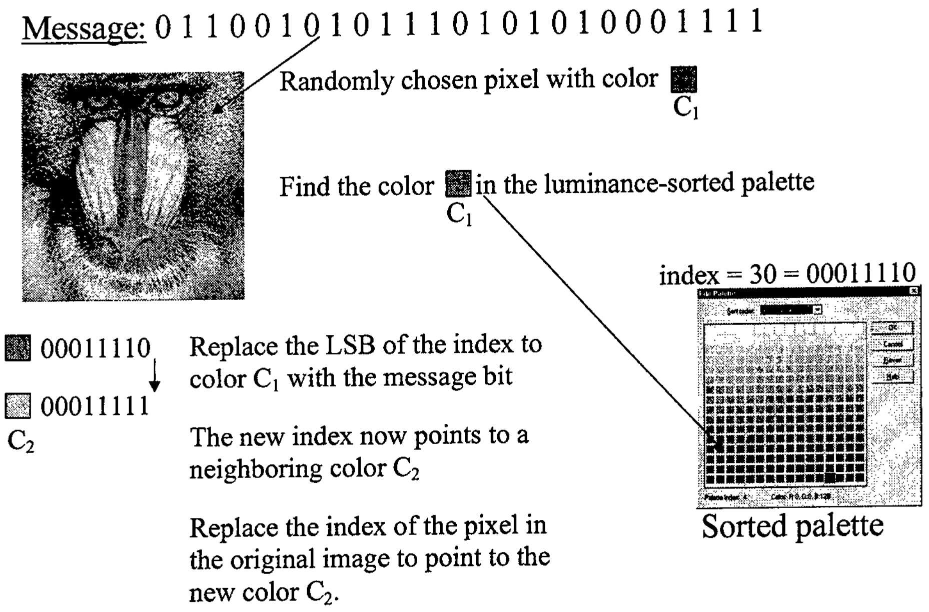
Other available software tools hide the message into the image data by decreasing the color depth of the GIF image to 128, 64, or 32 before the embedding commences. With this method, when the least significant bits of one, two, or three color channels are perturbed, the total number of newly created colors will be at most 256. [John98] points out, however, that the newly created palettes will have easily detectable groups of close colors. It is thus relatively easy to distinguish images with and without embedded messages.

Still other software tools hide the message into the image data in the form of parities. Two examples of this method are implemented in EzStego and in Secure Stego.

EzStego for Indexed Images

One of the most popular message hiding schemes for indexed images (i.e., GIF files) has been proposed by Romana Machado [Mach]. In her method called EzStego, the palette is first sorted by luminance. EzStego embeds the message in a binary form into the LSB of indices (pixels) pointing to the palette colors. Since the palette is sorted according to the colors, typically invisible changes will be introduced using this algorithm. Here are the steps:

1. Find the index of the pixel's RGB color in the sorted palette.
2. Get one bit from the binary message and replace the LSB of the index.
3. Find the new RGB color that the index now points to in the sorted palette.
4. Find the index of the new RGB color in the original palette.
5. Change the pixel to the index of the new RGB color.


Message: 0 1 1 0 0 1 0 1 0 1 1 1 0 1 0 1 0 1 0 0 0 1 1 1 1
Randomly chosen pixel with color ▨ C_1
Find the color ▨ in the luminance-sorted palette C_1
index = 30 = 00011110
▨ 00011110
▨ 00011111
C_2
Sorted palette

Replace the LSB of the index to color C_1 with the message bit

The new index now points to a neighboring color C_2

Replace the index of the pixel in the original image to point to the new color C_2.

Figure 7. Algorithm for EzStego.

Message recovery is simply achieved by collecting the LSBs of all indices in the image file. Of course, the method could be improved by injecting message bits into randomly selected pixels based on a pseudo-random number generator (PRNG) seeded with a secret key.

The EzStego algorithm is based on the premise that close colors in the luminance-ordered palette are close in the color space. However, since luminance is a linear combination of three colors *R*, *G*, and *B*, occasionally colors with similar luminance values may be relatively far from each other. For example, colors [6,98,233] and [233,6,98] have the same luminance but represent two completely different colors. Therefore, on occasion, EzStego will introduce large changes in color during message embedding.

Secure Stego for Indexed Images

To avoid the problem inherent to EzStego, [Frid99b] proposes to hide message bits into the parity bit of close colors. For the color of each pixel into which we embed message bits, we search the closest colors in the palette until we find a palette entry with the desired parity bit[10]. Since the parity bits of palette entries corresponding to real images are more or less randomly distributed, this will guarantee that we will never have to depart from the original color too much. This way, we avoid the problem of occasionally making large changes in color. [Frid99b] introduces four to five times less distortion as measured by root mean square error (RMSE) in the color space than EzStego.

Because the choice of the parity greatly influenced the amount of distortion introduced into the cover image, [Frid99e] explored the possibility of generating the optimal parity assignment. A rigorous proof is given that shows that the optimal parity assignment introduces -- on the average -- the least distortion into the image as measured by the RMSE. [Frid99e] also extends the optimal parity assignment for multiple-pixel embedding which embeds the message bit into q-tuples of pixels, where $q > 1$. Empirical data shows that when $q = 3$, one introduces half of the distortion that would have been embedded if $q = 1$.

It must be noted that RMSE was used in comparing EzStego with Secure Stego. This type of measure is accepted to be inadequate. It does not take into consideration the human visual system and its ability to recognize the distortion due to message embedding. However, in the absence of an acceptable metric, RMSE is used as the *de facto* metric.

3.2 Spectral Techniques

Spectral techniques are more robust than spatial techniques with respect to image processing operations. However, spectral techniques cannot be applied to all types of images. For example, spectral techniques only work with intensity and RGB images because of the large number of available intensities or colors. Indexed and binary images, on the other hand, are very intolerant to spectral data embedding techniques due to their low color depth.

[10] In [Frid99b], the parity bit of the color (R, G, B) was defined as R+G+B mod 2, although one can arbitrarily assign parities to the palette color.

Figure 8 illustrates the general procedure for frequency domain embedding. Upon applying a frequency transform to the data, a perceptual mask is computed that highlights perceptually significant regions in the spectrum that can support the watermark without affecting perceptual fidelity. The data to be embedded is then inserted into these regions.

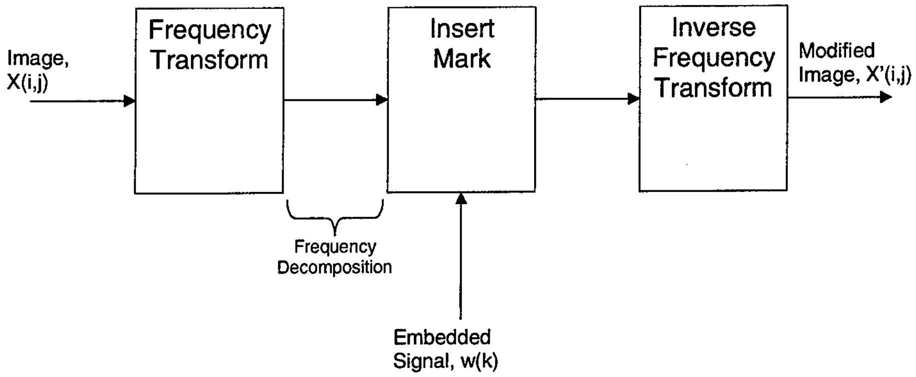


Figure 8. Data Embedding in the Frequency Domain.

In principle, any frequency domain transform can be used. Of the different possible transforms, DCT, FFT, and Wavelet transforms are the most popular. [Frid98a] proposes to use a key-dependent, random orthogonal transform basis instead of the publicly known transforms. The logic behind this choice is to make directed attacks to remove the embedded data much more difficult to conduct since the basis functions are unknown to the attacker. It is arguable, however, whether this increases the embedded data's robustness with respect to blind attacks.

[Ramk99] suggests the choice of the transform should depend on the robustness needed. For robustness against low frequency processing scenarios, the discrete cosine and wavelet transforms are appropriate choices since they are better energy compaction transforms. But if robustness against high frequency operations is required, one should use a transform with slightly inferior energy compaction properties like the Hartley transform.

This paper presents two popular spectral data embedding techniques. The first was introduced in [Cox95a]; the second presented in [Ruan98]. Since [Cox95a] and [Ruan98] are based on direct spread spectrum encoding, they possess both the advantages and disadvantages of spread spectrum techniques. They are extremely resistant to non-linear distortions of amplitude and to additive noise, yet are extremely intolerant to timing errors (loss of synchronization).

Synchronization is of utmost important during watermark extraction. If the original image is present (which is true for all non-oblivious techniques), resynchronization is relatively trivial [Ruan98]. If not, the problem becomes extremely difficult. If the watermark image is translated, rotated, and scaled, then synchronization necessitates a search over a four dimensional parameter space (X-offset, Y-offset, angle of rotation, and scaling factor). The search space grows larger if there is the possibility of shear and of changes in aspect ratio. As pointed out earlier, without having access to the original image, resistance to these geometric manipulations (e.g., translation, resizing, rotation, cropping, and warping) is still an unsolved issue.

Cox *et al*

Cox *et al* in [Cox95a] introduces a non-oblivious, private marking scheme that may employ perceptual masking. It uses a spread spectrum approach to spread one message bit across the entire image. It embeds a single-bit message into the image.

Watermark Encoding

Figure 9 shows the process for encoding an image with a spread spectrum watermark. Matlab code of the Cox *et al* data embedding encoder is listed in Section 8. The embedding process consists of the following:

1. Generate the DCT of the original image.

2. Generate the perceptual mask that identifies the significant perceptual components of the DCT. [Cox95a, 95b, 96a, 96b] indicated that there are many different methods for determining the most significant perceptual components and that this is an area for further research. [Giro89, Swan96c, Podi98] give other visual models to determine the most significant perceptual components into which to embed the data. The

method [Cox95a] opted to use the largest 1000 DCT components excluding the DC component.

3. Insert the watermark into the perceptual components of the DCT. Cox *et al* discussed the trades in choosing the length of the watermark. For instance, the length of the watermark affects its spread among the significant components of the images. In general, the magnitude of each watermark component has to be decreased as the length of the watermark increases to prevent noticeable image degradation. If an image is especially sensitive to changes in its DCT coefficients, then a longer watermark with smaller coefficients may be used without significant degradation of the watermark's robustness.

   Cox *et al* proposed using a watermark consisting of a sequence of real numbers. The length of the sequence is defined as the length of the watermark, in this case: 1000. Many variations for generating a watermark are possible. For example, one may tune the watermark for maximum robustness for a selected set of image corruption methods. Each watermark real number value was chosen independently using a random standard normal distribution.

   Each watermark value was inserted into a DCT component using the additive-multiplicative equation:

$$v_i' = v_i(1 + \alpha x_i)$$

where

x_i = the ith watermark value

v_i = the original DCT element value

v_i' = the modified DCT element value

α = the strength of the watermark

The watermark strength was set to $\alpha = 0.1$ in Cox *et al.*

4. Generate the watermarked image by taking the inverse of the modified DCT. Inverting the watermarked DCT produced the watermarked image.

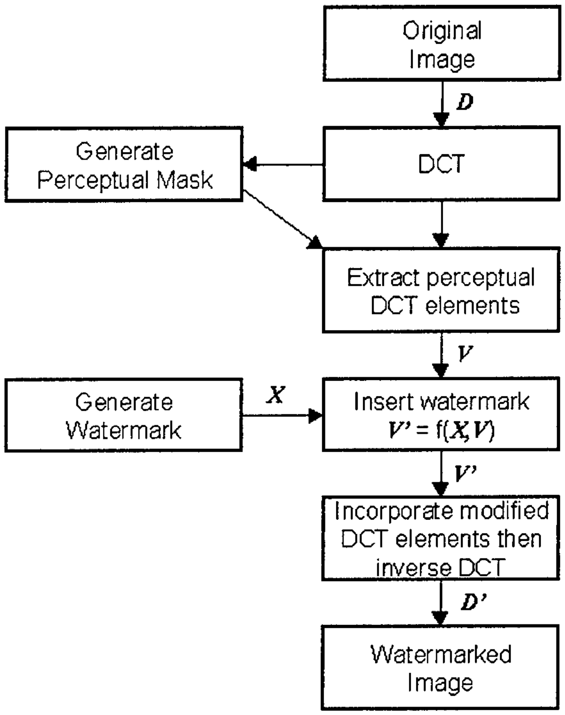


Figure 9. [Cox95a] Watermark Embedding Algorithm.

Watermark Detection

Cox *et al* recovers the watermark by explicitly computing the correlation between the (noise corrupted) watermark recovered from the image with a set of perfect watermarks stored in a database. This is a very robust technique for watermark recovery, but it is not very useful in practice because of the need for access to the database of perfect watermarks and of the large amount of computation required.

Figure 10 shows the watermark detection process developed by Cox *et al.* Matlab code of the Cox *et al* data embedding decoder is listed in Section 8. It consists of the following steps:

1. Compute the DCT of the recovered and of the original unwatermarked images.

2. Apply the perceptual mask to extract the DCT elements that contain the watermark. The perceptual mask used in the watermark insertion process identifies the DCT locations in the recovered image to look for watermark information. The same DCT locations are also extracted from the original images. The largest 1000 DCT components, excluding the DC component, is used in this implementation.

3. Rescale the DCT perceptual elements of the recovered image. The extracted perceptual elements from the recovered image are rescaled to match the vector length of the perceptual elements from the original image. This rescaling is an addition to Cox *et al* watermark detection process. It was introduced by Mike Hafer [Hafe97].

   Without the rescaling step, [Hafe97] found that the watermark detection process was sensitive to the range of the image intensity values. Two methods were investigated to reduce this sensitivity. The first method is to rescale the intensity values of image for the same minimum to maximum range as the original watermarked image. The second method rescales the length of the DCT perceptual vector of the recovered image to match that of the original image. The rational for method (2) is that the intensity range effects the absolute magnitudes of the DCT elements but not their relative magnitudes. Therefore, the length of vector V* before rescaling is affected by intensity range variations, but its direction is not.

   Rescaling the perceptual DCT elements to match the original unwatermarked image (instead of to the original watermarked image) insures that a watermark bias is not applied to the recovered image. This bias could have potentially increased the probability of a false positive. Both of these methods eliminate any watermark detector reading variation due to scaling alone.

4. Extract the watermark by using the inverse of the insertion equation. The watermark is extracted by using the inverse of the equation used to insert the watermark. The insertion equation was

$$v_i' = v_i(1 + \alpha x_i)$$

and its inverse is

$$x_i^* = (v_i^* - v_i)/(\alpha v_i)$$

where,

x_i^* = the ith recovered watermark value

v_i = the original DCT element value

v_i^* = the recovered DCT element value

α = the strength of the watermark

Recall, that the watermark strength used was $\alpha = 0.1$.

5. Measure the similarity of the extracted watermark against the original watermark. The similarity of the original water mark X and the recovered watermark X* was computed by Cox *et al* in their experiments using:

$$\mathrm{sim}(X, X^*) = (X^* \cdot X)/\mathrm{sqrt}(X^* \cdot X^*)$$

One can normalize sim(X, X*) was normalized for a maximum value of one by dividing sim(X, X*) by the Euclidean norm of the original watermark vector, X:

$$\mathrm{sim\_norm}(X, X^*) = \mathrm{sim}(X, X^*)/\mathrm{sqrt}(X \cdot X)$$

Normalization provides the advantage of knowing the maximum watermark detector reading for any watermark. This simplifies comparisons when different watermarks were involved. A 100% watermark hit would always generate a 1.0 output from the detector independent of the element values of the original watermark.

The detection is based on hypothesis testing and the existence of an *a priori* threshold. If sim(X,X*) or sim_norm(X,X*) is above a certain threshold, the watermark is said to be present in the image. Otherwise, the watermark is considered absent.

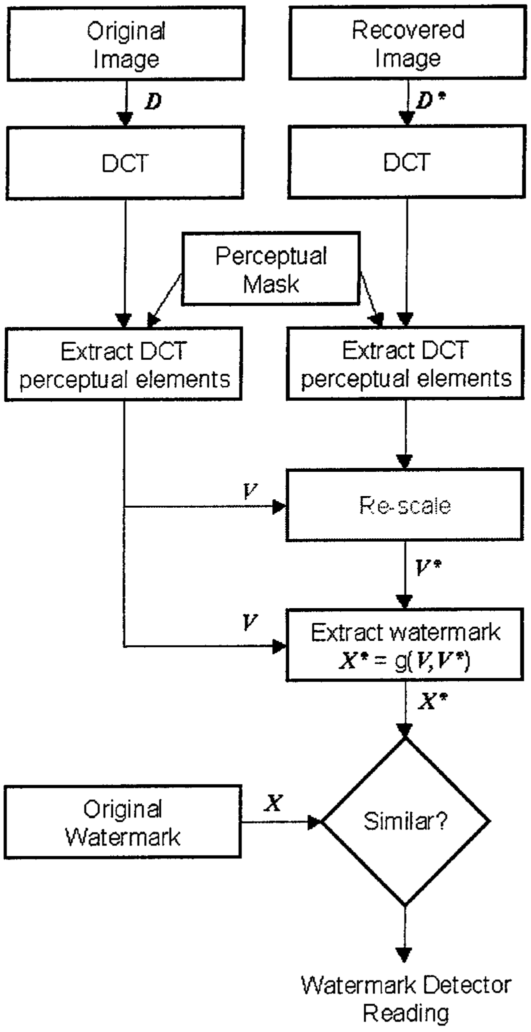


Figure 10. [Cox95a] Watermark Detection Algorithm.

Refer to [Cox95a, b, 96a, b] for results of the robustness testing for this technique.

Ó Ruanaidh

An elegant method for coding multiple bits into a spread spectrum signal has been described by Ó Ruanaidh [Ruan98] (a similar technique was proposed by Piva [Piva97]). It is similar to the technique proposed by Cox *et al*, since it is based on spread spectrum techniques. However, it does not require access to a database of watermarks nor does it need prior knowledge of which watermark to detect. The technique introduced by [Ruan98] is an oblivious data embedding scheme that may perceptual masking. It embeds a readable, multiple-bit message into the image.

The watermark is inserted by adding a noise-like signal to the middle frequencies of its DCT. The DCT coefficients are converted to a vector and the middle 30% (N_m frequencies) is chosen for marking. The information carried by the watermark consists of M symbols and each symbol s_i is represented using r bits, $1 \leq s_i \leq 2^r$. For each i, a sequence $\xi^{(i)}$ of pseudo-random numbers of length N_m+2^r uniformly distributed in [0,1] is generated. For each symbol a new sequence of pseudo-random numbers is generated. Symbol s is represented using the segment $\eta^{(i)} = \xi_s^{(i)}, \ldots,$ $\xi_{s+N_m-1}^{(i)}$ of consecutive N_m pseudo-random numbers. These pseudo-random vectors may be generated by any "good" pseudorandom number generator (e.g., Gold Codes, Kasami Codes, m-sequences, Legendre sequences, and perfect maps). The seed for the PRNG serves as the secret key. The message of M symbols is then represented as a summation

$$S_p = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \eta^{(i)} .$$

The spread spectrum signal S_p is approximately Gaussian with zero mean and unit standard deviation even for moderate values of M (e.g., $M \approx 10$). The signal S_p is further multiplied by a parameter γ(watermark strength / visibility) and added to the middle N_m DCT coefficients d_j. Again, the spatial masking model of Girod [Giro89] can be used to adjust γso that the double watermarked image is perceptually identical to the original image. The value of γ= 13 works well for most images. The amplitude of the combined watermark is typically in the range [–20,20] with an average rms of 5 gray levels. In [Ruan98], the watermark was repeatedly embedded in blocks of 128×128 pixels. Matlab code of the [Ruan98] data embedding encoder is listed in Section 9.

The detection of the message consisting of M symbols proceeds by first transforming the image using a DCT and extracting the middle N_m DCT coefficients. The secret key is used to generate M pseudo-random sequences of length N_m+2^r needed for coding the message symbols. For each sequence, all 2^r segments of length N_m are correlated with the middle N_m DCT coefficients. The largest value of the correlation determines the encoded symbol. Matlab code of the [Ruan98] data embedding encoder is listed in Section 9.

This watermarking scheme exhibits very impressive robustness properties with respect to many image processing operations. Brightness/contrast adjustment, gamma correction, histogram operations, dithering, sharpening, noise adding, and high-pass filters leave the watermark almost untouched. The watermark is also fairly robust to lossy JPEG compression. Depending on the watermark strength, the message can supposedly be extracted untouched after JPEG compression with 15% quality. Low pass filtering, mosaic filter, and median rapidly deteriorate the watermark especially when applied iteratively several times.

Refer to [Ruan98] for the results of robustness testing for this technique.

4 Limitations of Data Embedding Schemes

Given a data embedding scheme, an unscrupulous person can devise a distortion that removes or destroys the embedded data without severely degrading the quality of the image. The attacker has the advantage here. It is inconceivable that a data embedding technique is designed to be robust with respect to all possible attacks. This section presents three general types of attacks: robustness attacks, presentation attacks, and interpretation attacks.

4.1 Robustness attacks

Robustness attacks aim to diminish or remove embedded data from the stego object. A basic attack of this nature would be to lossy compress an image in the hope that the resulting image would no longer have the embedded data. Pure steganographic techniques that trade-off robustness for undetectability are very susceptible to common signal processing operations.

There are a number of data embedding techniques that have been developed for digital watermarking which are very robust with respect to JPEG compression, additive Gaussian noise, low pass filtering, scaling, cropping, brightness/contrast enhancement, global rotation, and cropping. Even so, these techniques are based on spread spectrum methodologies. And although spread spectrum signals are very robust to amplitude distortions and to noise adding, they are very prone to synchronization errors. Synchronization is very important and simple systems fail to recover this synchronization properly.

4.2 Presentation attacks

Presentation attacks modify the content of the stego object such that the detector cannot find the embedded data. This paper discussed earlier the Stirmark attack, which is a very general attack. Stirmark introduces sub-perceptual distortions that render the embedded data unreadable. Another general presentation attack is the mosaic attack.

The mosaic attack was motivated by the creatios of autonomous agents that will search the Internet for images that contain certain embedded information. An example of this is MarcSpider by Digimarc Corporation [Digi99]. The MarcSpider service searches the public Web for images containing digital watermarks and produces reports on where and when such images are found. Autonomous agents similar to the MarcSpider allows Web content developers, photographers, stock photography agencies and publishers of entertainment, sports and news images to track their works on the Web.

The mosaic attack consists of chopping an image into a number of smaller sub-images. These smaller images are tiled on a web page such that a web browser would render the juxtaposed sub-images such that they appear identical to the original image. [Peti98] argues that this attack is quite general; all marking schemes require the marked image to have some minimal size (one cannot securely and robustly embed a meaningful mark in just one pixel). If an image is divided into images smaller than this minimal size, the embedded data can no longer be recovered.

4.3 Interpretation attacks

Interpretation attacks devise situations which prevent the assertion of the embedded data. In a digital watermarking architecture, a simple interpretation attack will be to insert a second watermark into a previously watermarked image. If a naïve embedding scheme was used, there would be no intrinsic way of determining which of the two watermarks was added first. The invertibility problem presented by [Crav99] -- which is also known as the IBM attack -- is an example of an interpretation attack. Timestamping and notarization, as they are performed by the U.S. Copyright Office [USCo], may prevent attacks of this type.

5 Applications

There are four clusters of applications for data embedding. One application uses data embedding techniques to convey ownership information (source authentication). Another application uses digital watermarking techniques to verify that the object content has not been changed (content authentication). A third application uses steganography to convey object-specific information (e.g., captions) to a community of willing recipients. The last application is the use of steganographic techniques to convey a secret message while concealing the very presence of the secret message.

5.1 Source Authentication

In the business environment, applications that convey ownership information are often desired by organizations that own the copyrights to digital media objects and that license them. [Mint99] describes a typical image application as follows: the content owner provides an image that is published by the recipient; in exchange, the owner receives a royalty for the use of the image. A concern of the owner is that the publisher may, intentionally or unintentionally, neglect to pay the royalty. To deter such misappropriation, the owner may wish to place ownership information in the image.

Traditionally, handwritten signatures have been used as proof of authorship of, or at least agreement with, the contents of a document because of the following assumptions made on the signatures[11] [Schn96]:

1. The signature is *authentic*. The signature convinces the document's recipient that the signer deliberately signed the document.
2. The signature is *unforgeable*. The signature is proof that the signer, and no one else, deliberately signed the document.
3. The signature is *not reusable*. The signature is part of the document; an unscrupulous person cannot move the signature to a different document.
4. The signed document is *unalterable*. After the document is signed, it cannot be altered.
5. The signature *cannot be repudiated*. The signature and the document are physical things. The signer cannot later claim that he or she did not sign the document.

With digital documents, we would like to use a concept similar to handwritten signatures. However, there are problems due to the nature of digital media. First, computer files are trivial to copy -- it is easy to cut and paste a valid signature from one document to another. The mere presence of such a signature means nothing. Second, computer files are easy to modify after they are signed, without leaving any evidence of modification.

[11] [Schn96] points out that in reality, none of these assumptions is completely true. Signatures can be forged; signatures can be lifted from one piece of paper and moved to another; and documents can be altered after signing. However, Schneier notes we are willing to live with these problems because of the difficulty in cheating and the risk and penalty associated with detection.

To overcome these potential weaknesses, digital signature protocols were developed [Schn96]. However, a weakness of using digital signatures is that the signature is separable from the document. An unscrupulous individual can throw away the original signed hash, generate his own hash with his own private key, and claim ownership to the document. To avoid this situation, data embedding techniques can be used.

Figure 11 illustrates the source authentication scenario using a data embedding protocol. The author of a digital image wants to "sign" the image so that no one else can attribute the authorship of the image to himself. The signature can be appended to the image file, be visibly imprinted on the image, or be protected by a digital signature. However, these signing methods have their drawbacks. If the signature is appended to the image file, an attacker can simply remove or replace it with the attacker's own signature. If the signature is visibly imprinted on the image, an attacker can simply crop it out. Cryptographic digital signatures offer no protection because they can be simply replaced. If the author's signature is embedded into the image, a recipient can determine who owns the image by detecting the embedded signature. In addition, if the signature is embedded into the image, it is hoped that the attacker has a more difficult job of removing the signature without severely degrading the image.

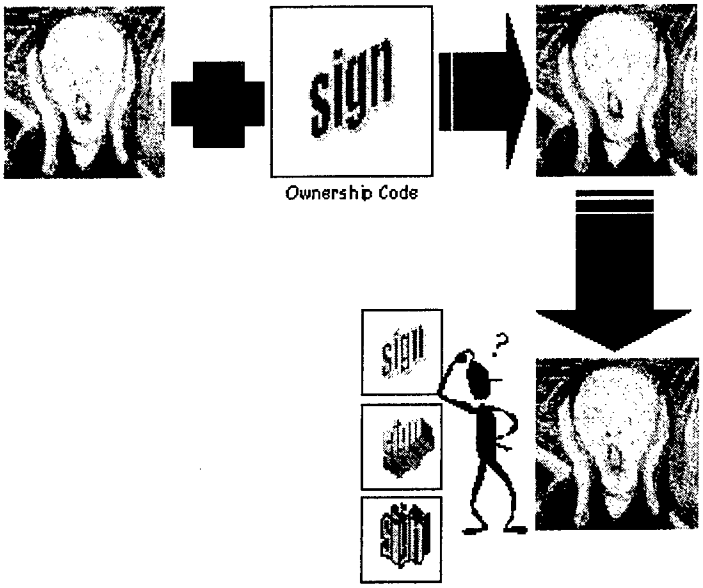


Figure 11. Source Authentication.

The ownership information may identify either the owner or the recipient [Mint99]. If the watermark identifies the owner, the owner might subsequently scan suspect published material to determine whether a printed image contained his watermark; the owner would consider its presence to be evidence of his ownership. If the watermark contained the recipient's identification, the owner might subsequently scan the published material to determine who received the material. If the image had been used without the payment of royalties, the owner might wish to cease doing business with the recipient.

In military applications, instead of embedding ownership information, one can embed source information. In a typical image application, a reconnaissance sensor can tag its products with its ownership. Since the information is embedded into the image data, it is inseparable from the image. One will always be able to retrieve information pertaining to the source of the imagery. This is more advantageous than having this type of information stored in a header field or in a separate file: header information can always be overwritten; links to the separate file can always be broken.

For a data embedding scheme for source authentication to be effective, the embedded information (i.e., the watermark) must remain in the media object and must be reliably detectable from the published product. This is a considerable challenge. A digital image, for example, can undergo numerous transformations in its lifetime. Sharpening, contrast enhancement, color correction, filtering, and compression are just examples of image manipulation operations an image can undergo. A watermark must survive these various image processing operations. In addition to simple image processing operations, a watermark must survive geometric distortions (e.g., rotation, cropping, re-scaling). Moreover, it must also be taken into consideration that there is an economic incentive for unscrupulous people to remove watermarks. Thus, an ownership watermark must also be robust with respect to directed attacks against the watermark. Watermarks that withstand intentional or unintentional attacks are considered robust watermarks. Previous works are [Cox95a], [Hart97c], [Kund97], [Ruan98].

Aside from the robustness issue, a reliable source authentication architecture must exist to support the source authentication scheme. [Adel99] proposes an architecture that might provide the support needed for a source authentication / copyright protection protocol.

A generic copyright protection protocol will have two mechanisms: a registration mechanism and a arbitration mechanism. The registration mechanism (Figure 12) will accept a digital object (V_A) and a digital identity (ID_A) from a principal, say Alice. With these objects, it will determine whether the digital object had been previously registered. If the digital object had been registered, the registration mechanism will abort. If not, the registration mechanism will issue Alice a proof of ownership and the watermarked version of V_A.

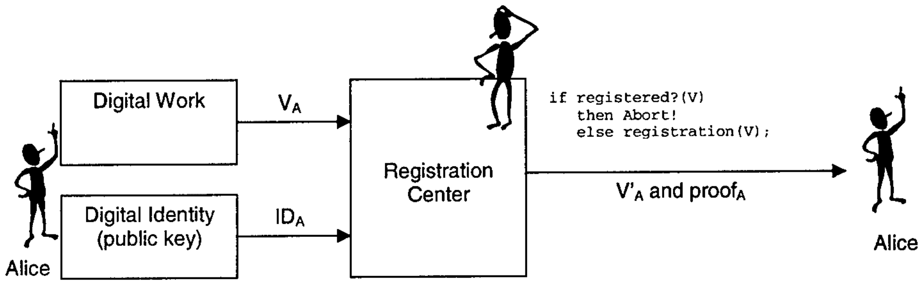


Figure 12. Generic Registration Mechanism.

In an ownership dispute, the arbitration mechanism (Figure 13) will require the disputing parties to present their digital identities and their proofs of ownership, as well as the digital work under dispute. The registration center will have to determine which of the disputing parties is the rightful owner of the digital work. The registration center must also check whether the digital work belongs to a party not involved in the arbitration. If the registration mechanism competently performed its task, the arbitration mechanism can simply verify the proofs of ownership presented to it and search its database for the rightful owner.

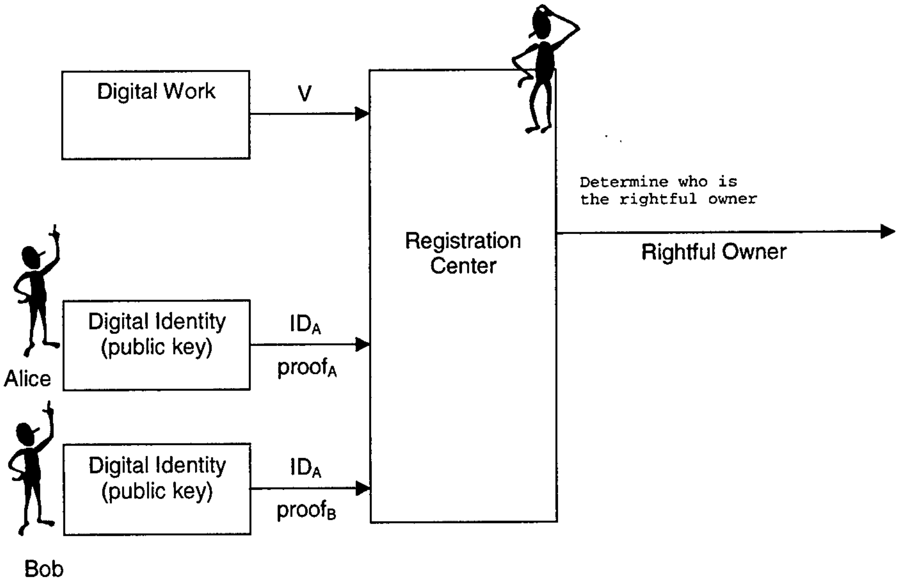


Figure 13. Generic Arbitration Mechanism.

However, the registration mechanism's responsibility is more complex than presented here. In determining whether a digital work has been previously registered, similarity tests must be performed. Is the digital object being registered similar to a previously registered object? The similarity tests must consider an equivalence relationship and not an equality relationship. For example, the similarity test should not be deceived by scaling-rotation attacks or by brightness / contrast enhancements. The nature of the equivalence test is an open issue and is one of the most difficult problems in this application. The performance of the registration and arbitration mechanism must also be taken into consideration. The time it would take to search the database for similar, registered objects must be acceptable.

The registration center must also be trusted and held accountable for its actions. As pointed out by [Adel99], a dishonest center could

1. refuse registration, after obtaining the original work, and claim that a similar work has already been registered,
2. issue a wrong ownership certificate, i.e., a certificate not suitable for ownership proofs,
3. collude with a unscrupulous party and issue him an ownership certificate for an previously registered object, and
4. return an incorrect result during an ownership dispute

The copyright protection architecture is similar to the existing United States Copyright Office. However, without a robust and comprehensive similarity test, it is debatable whether such an architecture can be automated. Nevertheless, the use of robust digital watermarks can still be used as an initial check for ownership. For more information on the U.S. Copyright Office, the reader is referred to http://lcweb.loc.gov/copyright/.

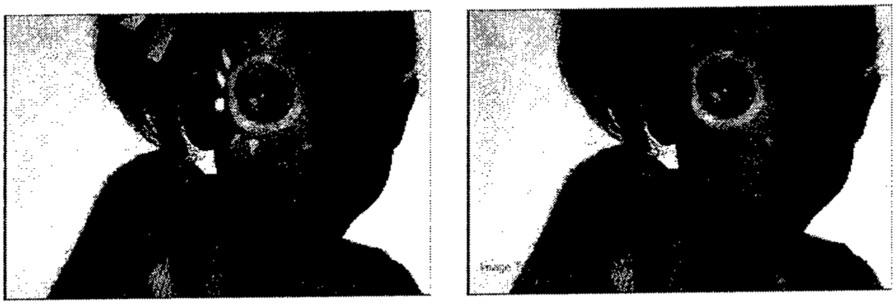
5.2 Content Authentication

The second application area uses watermarks to determine whether a media object has been altered since some earlier time when it was watermarked. Current authentication techniques are based on cryptographic principles and digital signatures. The advantages of using a digital signature protocol are twofold. First, it is computationally easy to compute a hash value. This minimizes the overhead associated with signing documents. Second, the signature is sensitive to any modification to the post-transmission document. Digital signatures are good for *complete authentication* -- the data considered as untouchable [Lin99]. These techniques protect every single bit of the content and do not allow any form of manipulation. As such, to be considered authentic, the data under test has to be equal to the original one.

In real applications, this may not be practical. In several situations, digital images may be lossy compressed and may be digitally enhanced and manipulated. Moreover, they may simply be converted from one graphical file format to another. The operations do not affect the semantic meaning of the image. Because the significance of digital images is based on their content, and not on their bits, manipulation on the bits that do not change the meaning of the content should not affect an image's authenticity. On the other hand, manipulations that add or remove features from an image should be detected and possibly localized while validating the remaining areas of the image.

Since the meaning of digital images is based on its content, one can modify the image bits to embed some codes, i.e., watermarks, without changing the meaning of its content. Because the watermarks are embedded in the data content, once the data is manipulated, these watermarks will also be modified such that the authenticator can examine them to verify the integrity of the data. In a typical image application, an image is watermarked at the time it is loaded into a digital library. Later, the watermark is extracted. If the extracted watermark matches the inserted watermark, the object is judged to be unchanged; if it does not match the inserted watermark, the image is judged to have been altered. An inspection of the extracted watermark and its difference from the inserted watermark can be used to reveal where the alterations have occurred. [Frid99c] has also shown that it is possible to store information about the image in the image itself. If the image has been modified, one can localize where the image had been altered and reconstruct the pre-modified image.

Adequate content authentication schemes must be able to distinguish between malicious and non-malicious attacks on the image. Malicious attacks are attacks that modify the content of an image. These attacks include manipulating, deleting, or adding features in an image. Non-malicious attacks modify the pixel values of an image, but do not modify its contents. Examples of non-malicious attacks are adjustment of brightness or contrast, lossy compression, or filtering. Research conducted using data embedding techniques for content authentication use fragile watermarking schemes. These fragile watermarking schemes are robust to certain image processing operations, but will be significantly altered or destroyed if the image is altered. Previous works are [Walt95], [Lin97, 98a, 99], [Wu98], [Yeun97], [Frid98d, 98e, 99c], and [Bald99].



Original Image

Tampered Image

Figure 14. Content Authentication Scenario.

Figure 14 illustrate the need for content authentication techniques. With the advent of digital images, the concept of an "original" image becomes blurred. Unlike traditional images and their chemical-based processing, digital images do not have the luxury of having negatives to verify content authenticity. However, if the digital image was specially prepared, unauthorized modifications to an image can be detected. Tamper detection seals were embedding in the least significant bits of the original image in Figure 14 in accordance with the scheme proposed in [Bald99]. Baldoza *et al* extends the authentication scheme proposed in [Walt95] by subdividing the entire image into sub-blocks and ultimately protecting each sub-block by introducing the walk-dependent checksum. Because the original image is divided into sub-images, one can localize where the image had been altered. Figure 15 demonstrates the capability of detecting and localizing unauthorized modifications to the image.

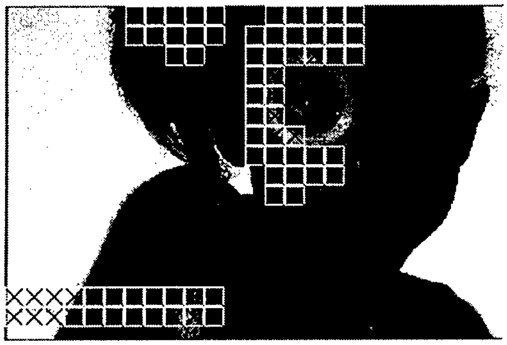


Figure 15. Detection of Tampering in the Content Authentication Scenario.

Figure 16 illustrates potential instances in an imagery collection and dissemination process wherein the integrity of an image can be compromised. At Point A, the source of the imagery can be altered if an adversary has the ability to intercept communication between a friendly reconnaissance platform and its support communication relay. The hostile imagery platform can either degrade the original imagery or substitute it for a downgraded and possibly altered image. At Point B, attacks can be taken against the imagery at several instances. For example, the imagery can be attacked while it resides on the database prior to authorized exploitation. In addition, the imagery can be inadvertently modified by an image analyst. In either case, the intelligence gathered by exploiting an altered image or by faulty exploitation may have grave consequences.

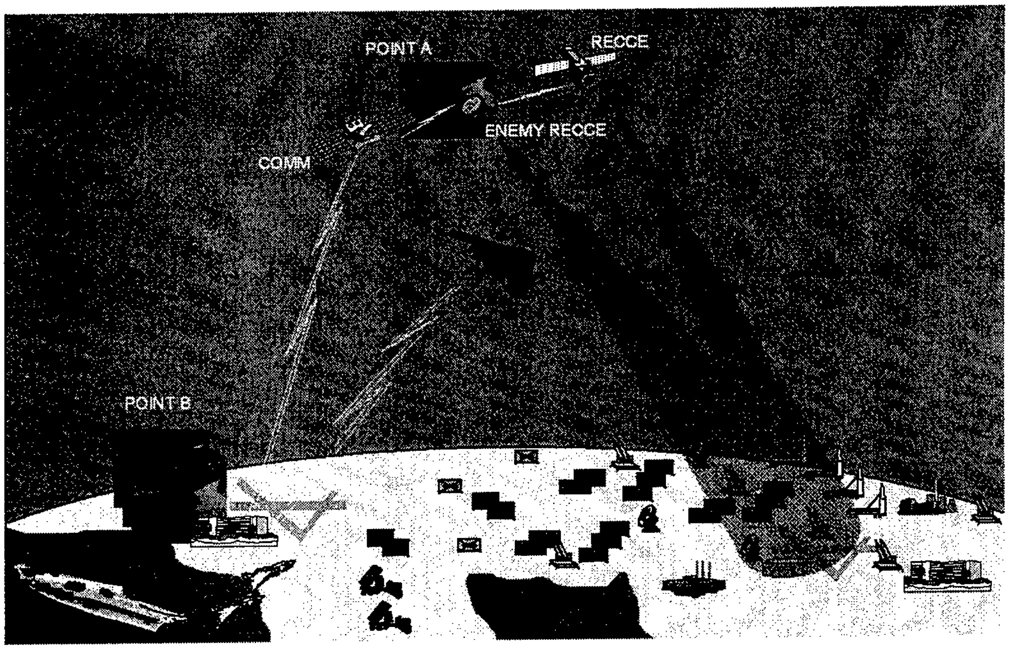


Figure 16. Potential Security Breaches in the Collection and Dissemination Architecture.

Cryptographic techniques can be used to safeguard collected imagery, but only while the image remains in ciphertext form. Once the cryptographic wrapper is removed, the images are susceptible to malicious and inadvertent modifications. This problem was discussed earlier in the section dealing with source authentication using digital signatures.

The weakness of using this cryptographic protocol is not that the adversary can throw away the original signed hash, generate his own hash with his own private key, and claim ownership to the image. Instead, the weakness is that once the image has been authenticated, the signed hash may be disassociated with the image -- through negligence or through malicious intent. If this occurs, subsequent authentications are not possible. Any modification to the image once the signed hash is disassociated with it is undetectable. This weakness can be addressed by using data embedding techniques to embed tamper-detection seals into the image. Ideally, these seals will break when the image has been altered.

Several different schemes that use data embedding techniques can be used, each providing certain advantages. If any modification to the image is to be detected, fragile embedding techniques that "break" at the slightest perturbations should be used. If non-malicious modifications should not flag an image as being altered, techniques that are more robust should be used. Therefore, images that had undergone processing operations such as lossy compression, blurring, sharpening, and contrast/brightness enhancement should pass the authentication assessment. However, images wherein which features have been removed (e.g., a surface-to-air missile site had been cropped out) or modified (e.g., a face had been replaced) should fail the authentication process.

One image processing operation that should be considered non-malicious is ortho-rectification. Ortho-rectification is a complex operation involving subtle mathematical calibrations to represent the curved surface of the earth as a flat map. Although this operation does not change the content of the image, the non-linear geometric distortions introduced into the image currently renders any current data embedding technique impotent. Unless the non-linear distortions can be reversed, any embedded data is rendered unreadable by this operation. A data embedding technique that is robust to ortho-rectification must be developed before data can be reliably embedded into satellite and aerial imagery[12].

5.3 Captioning

Captioning applications are the friendliest environment for data embedding techniques. In captioning applications, both the content owners and the recipients desire that the information be conveyed – there is no economic incentive for unscrupulous people to remove the embedded data. In addition, it does not matter whether the embedded data can be detected. A typical image application has a digital photo agency embedding information regarding the name of the owning agency and an image identification number. When the image is published, the publisher knows whom to contact for permission. In addition, the photographer can embed aperture, shutter speed, and subject information to capture the conditions in which the photo was taken.

[12] Incidentally, any data embedding technique that is robust with respect to ortho-rectification will also be robust to the StirMark attack developed by F. Petitcolas [Peti98b]. This is the "holy grail" of commercial watermarking techniques -- an oblivious data embedding technique robust with respect to geometric deformations.

Traditional captioning techniques for digital images simply attach the caption to the image via a field in the file format or a separate, readable file. If security was an issue, the caption is encrypted before it is attached to the image. The main weakness to the simple attachment of the information is that the link between the image and the caption can easily be severed. In certain applications, this loss of information would not be beneficial for the consumer. To preclude the loss of the captioning information, one is motivated to embed the data within the image so that they become inseparable.

Captioning applications include movie dubbing in multiple languages, subtitles, and tracking the use of the data (history file). For example, one copy of a movie can be distributed with subtitles in several languages. The VCR, DVD player, TV set, or other video device can access and decode the additional text (subtitles) in real time from each frame, and display it on the TV screen. Although this could be arranged by appending information rather than invisibly embedding it, bandwidth requirements and necessary format changes may not allow us to do so.

Figure 17 illustrates the captioning application. Information regarding the image is embedded into the image. This information is available later for anyone to retrieve.

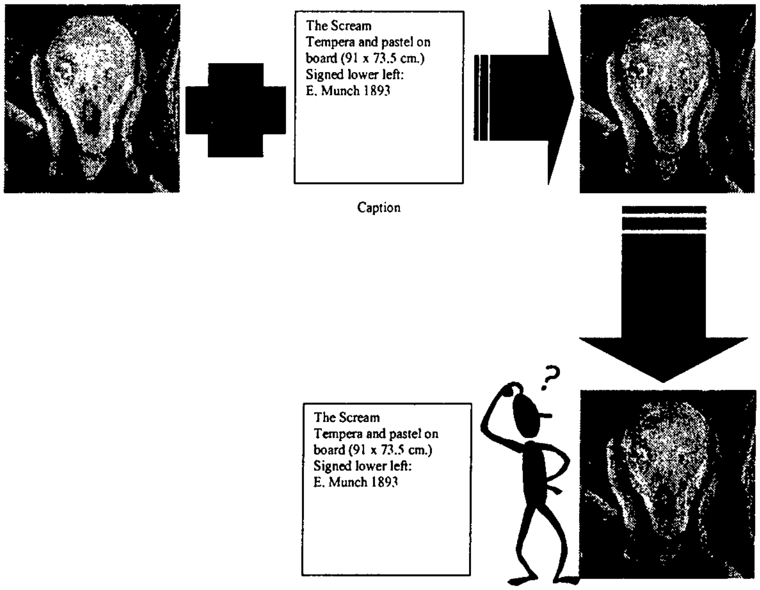


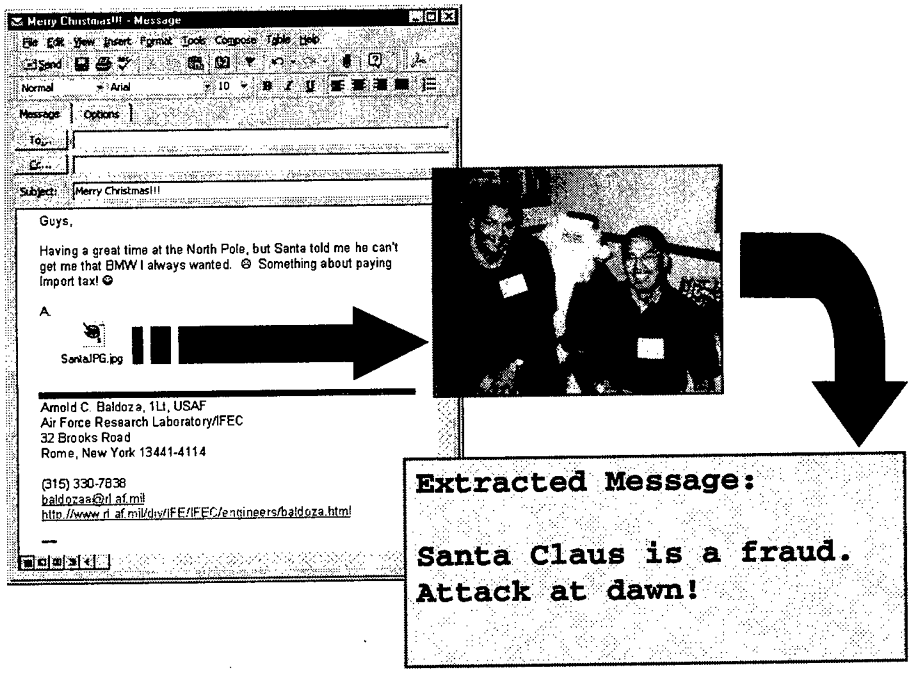
Figure 17. Captioning Application.

In military applications, one can embed metadata. In a typical image application, a reconnaissance sensor provides an image that will be exploited by the image analysts. Information regarding the sensor (e.g., sensor type) or its image acquisition parameters (e.g., longitude and latitude, slant angle) can be embedded into the image for subsequent retrieval. Again, this is more advantageous than having this type of information stored in a header field or in a separate file: header information can always be overwritten; links to the separate file can always be broken. A possible weakness of embedding the metadata is the upper limit on capacity – just how much data can be embedded into the image. In addition, there is a processing penalty associated with extracting the data.

In the captioning application, original images are not available for message extraction. Although the recipients desire to extract the embedded information, some robustness is required. Captions should survive unintended, non-malicious attacks (such as compression), but they are not required to survive directed attacks intended to remove the captions. As mentioned earlier, the consumer is not motivated to remove the hidden information since the hidden information is beneficial for the consumer. Performance, however, may be an issue -- if the embedded information must be recovered in real time (e.g., subtitles in a movie), fast detection is necessary.

5.4 Covert Communications

In lieu of embedding ownership codes or captions into an image, one can embed any bitstream into a digital image. This bitstream would be retrievable at a future time by either the embedder or an intended recipient. A typical scenario in a military setting would be a covert agent in a hostile environment embedding secret information within an innocuous-looking photograph. The agent will then transmit the photograph, along with an appropriate innocent-looking email message over the Internet. If the secret message was embedded properly, an email message along with its attached photograph will not arouse any suspicion. Any person (or program) scanning message traffic would only see the stego object – the innocent-looking picture and its message – and not the secret message. Figure 18 demonstrates this application. Attached to the email is a group photo into which the secret message had been embedded. Only with the correct steganographic key (and possibly a separate cryptographic key) will one be able to extract the embedded message.


Merry Christmas!!! - Message
File Edit View Insert Format Tools Compose Table Help
Send
Normal
Arial
10
Message
Options
To...
Cc...

Subject: Merry Christmas!!!

Guys,

Having a great time at the North Pole, but Santa told me he can't get me that BMW I always wanted. ☹ Something about paying Import tax! ☺

A.

Arnold C. Baldoza, 1Lt, USAF
Air Force Research Laboratory/IFEC
32 Brooks Road
Rome, New York 13441-4114

(315) 330-7838
baldozaa@rl.af.mil
http://www.rl.af.mil/div/IFE/IFEC/engineers/baldoza.html

—

Extracted Message:

Santa Claus is a fraud.
Attack at dawn!

Figure 18. Covert Communication Application -- Embedding the Secret Message in Innocent-Looking Email Messages.

The advantage steganographic techniques have over simple cryptography are obvious. Encryption simply transforms a message into something unintelligible. Since nothing is done to hide communication channel, the message traffic between the transmitter and the receiver remains observable. An eavesdropper can see the message traffic, but simply cannot understand it. With steganography, the secret message is hidden into innocuous messages. An eavesdropper can read the innocuous message traffic, but cannot detect the communication of the secret messages. For a secure communication channel, it is advisable to use both cryptographic and steganographic techniques. The steganographic wrapper will decrease the chances of the encrypted message traffic from being detected; the cryptographic wrapper will decrease the probability of the message from being read if the covert channel is discovered.

The capability of embedding messages in digital images has its dangers. An effective method will embed data into the images without introducing statistically detectable artifacts. It is not surprising that law enforcement agencies are attempting to develop algorithms that will detect the presence of messages in multimedia objects. In a military setting, there is a grave possibility that someone may have concealed classified data in declassified digital imagery [Kura92].

The most important requirement for covert communications is that the presence of the hidden message be visually imperceivable and statistically undetectable. Images with and without secret messages should appear identical to all possible statistical tests that can be carried out. In the covert communication application, detection of the presence of a hidden message is equated to the failure of the steganographic technique, even if detection does not imply the ability to read the message. Undetectability is of utmost importance especially in a military scenario wherein the presence and activities of covert agents must be kept secret.

For the embedded message to be visually imperceivable and statistically undetectable, it is of paramount importance to know as much about the statistical properties of the source from which cover images are being drawn. For example, if the images are scanned photographs, there will be stronger correlation in the horizontal direction than in the perpendicular direction [Frid98c].

[Fran99] attempts to create noise models of digital scanners in order to create secure steganographic techniques. If the images are taken using a digital CCD camera, the noise will again have certain specific properties induced by the CCD element and the specific data readout[13]. In either case, the data hiding scheme must respect all known statistical properties of the image source and produce images that cannot be distinguished from images that do not contain any messages. [Aura96] offers a discussion regarding the importance of respecting the statistical properties of the image source. In his paper, Aura successfully detected messages embedded using LSB encoding using statistical measures. In addition, [John98, 98b, 98c, West99] give a variety of methods of detecting data embedded in images.

Another important issue is the capacity of the communication channel. Capacity is also related to detectability. Obviously, the longer the message, the larger the modification of the cover image and the higher the probability that the modifications can be statistically detected. It is clear that one can embed one bit of information into one frame of a digital video without any consideration to noise models. Such communication scheme would however lead to an impractical and low communication bandwidth. The challenge is to embed as much information as possible while staying compatible with the image noise model. [Aura96] determined that replacing five percent of the LSBs with pseudo-random bits does not significantly change the statistical properties of an image. However, increasing the capacity such that 100% of the LSBs were replaced by pseudo-random bits renders the message detectable.

The last important requirement is that it must not be possible to detect the hidden message without the original image. If the cover object is available, it is trivial to detect embedded messages. Sometimes it may be possible to agree on certain image database from which cover images are drawn (without repetition!) but this obviously limits the applicability of the technique.

[13] In a recent technical exchange with Eastman Kodak, it was learned that images taken with digital cameras are pre-processed before being saved onto the storage device. Therefore, the

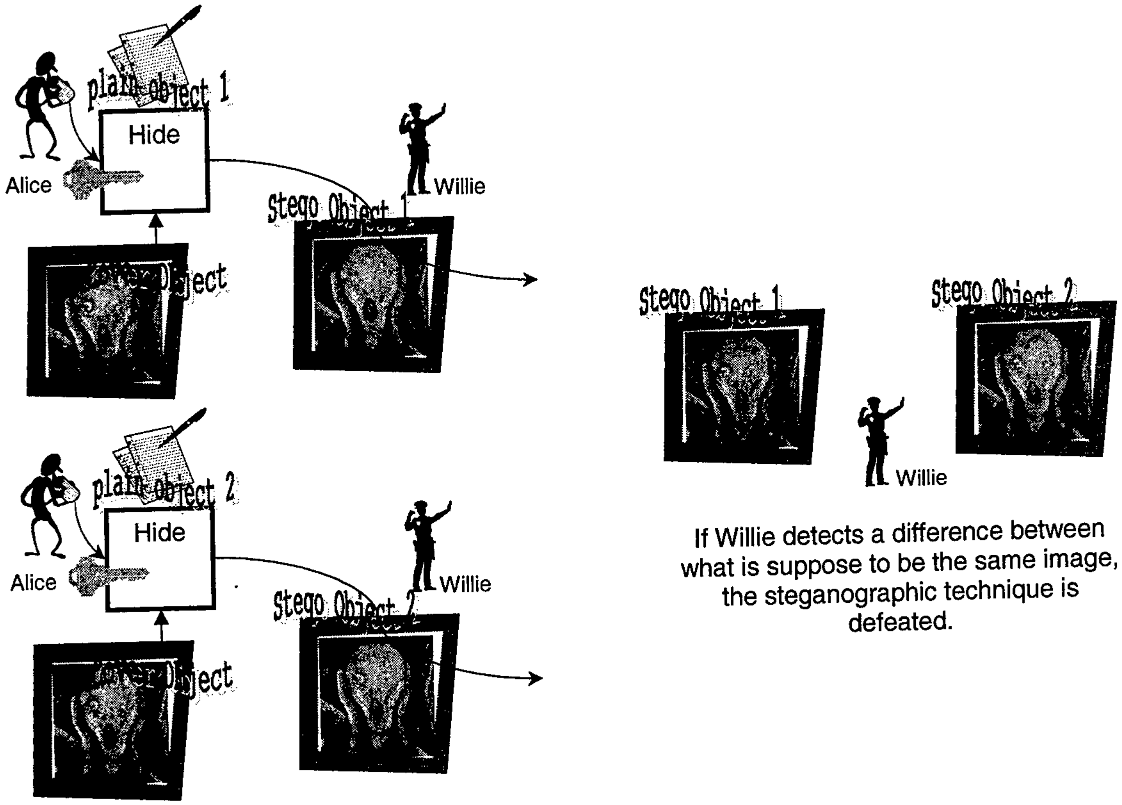
The data embedding architecture that would support special operation forces is relatively simple. In fact, the Prisoner's Problem epitomizes the scenario covert agents routinely find themselves in. The two principals -- Alice and Bob -- are the transmitter and receiver on the communication channel. Their communication channel is provided using email through the Internet and is susceptible to eavesdropping by Willie, the warden. If Willie detects any encrypted or suspicious message, he will incarcerate Alice or do worse. Therefore, Alice and Bob must find some way of concealing their covert messages in an innocuous-looking exchange of public messages.

Due to the ubiquitous nature of the Internet and of the growing popularity of digital cameras, a covert operator -- armed with a computer and a digital camera and provided with Internet access -- can easily blend into the general native populace. The operator can then transmit messages over the Internet either through email (via the attachment of a digital image) or through an Internet site (wherein the receiver can download the image). Steganographic techniques for indexed images or spectral techniques for JPEG images can provide the necessary covert channel of communication. Raw, 8- or 24-bit images are not recommended as cover objects in this type of scenario. Transmitting large 24-bit image files, for example, instead of their smaller JPEG equivalents may arouse suspicion. Furthermore, the posting raw images on web sites are rare, as most Internet browsers do not support them.

Undetectability is paramount. The uncovering of a covert agent carries great implications ranging from diplomatic embarrassment to the loss of lives. High capacity is also important. An increase in the volume of message traffic between two individuals, in itself, may not be cause for suspicion. However, the transmission of numerous messages quickly depletes the database of possible cover images. As discussed earlier, one should not even contemplate using the same cover object twice. Moreover, the original cover image should not remain available after the

noise model of these devices should take into consideration the specific properties of the CCD element, as well as the pre-processing algorithms.

embedding process. If one of these requirements are violated, the warden could perform a simple differential analysis between the two different stego objects and determine the presence of the covert message (stego* attack). Figure 19 illustrates the architecture of this attack. Differential analysis between the first and second stego object can be performed by simply subtracting the two objects pixel-wise. If one pixel in the first image is not equal to its corresponding pixel in the other object, Willie the warden will be alerted to the existence of a covert communication channel.



If Willie detects a difference between what is suppose to be the same image, the steganographic technique is defeated.

Figure 19. Stego* Attack.

Because one does not want the original cover object to be available the creation of the stego object, it would be advantageous to be able to embed the message at the source (e.g., on the digital camera). This would preclude the need of a computer to perform the message embedding. More importantly, it would also eliminate the existence of the original cover object. If the digital camera, for example, were to fall into the hands of the warden, the warden would be able to only view the stego objects.

Embedding at the source, however, carries with it potentially dangerous implications. Take the case where the source is a digital camera. One obvious requirement is that the steganographic technique embedded into the camera works on and outputs the same image formats supported by the camera. If the camera, for example, produces JPEG and/or (un)compressed TIFF images, then the steganographic technique must have as input and output JPEG and/or (un)compressed TIFF images. Steganographic techniques for indexed images cannot be used in this digital camera.

More importantly, the steganographic technique implemented on the camera must not violate the noise model of the digital camera. If the warden takes possession of the camera, he will be able to construct the camera's noise model using numerous subsequent photos taken by the camera. With this noise model, the warden will be able to test whether or not the images on the camera have the same noise characteristics. A single image that violates the noise model will signal the detection of an embedded message. It does not matter if the warden can read the embedded message -- the detection of the embedded message defeats the steganographic technique.

Implementation of the steganographic software/hardware on the digital camera is also difficult. If the warden has access to the digital camera, the steganographic software/hardware will have to be undetectable has well. If questioned, the owner would have to explain why his camera -- which is supposed to be commercially available -- has dedicated steganographic modules. This problem is compounded if a cryptographic system is also embedded in the camera. If steganographic hardware and software are commercially available and are recognized as authorized add-ons to the camera, the owner may be required to surrender his steganographic keys. The use of a multi-level steganographic file system, similar to that proposed by [McDo99], may have to be implemented.

Aside from implementation concerns, key management issues have to be addressed. Storing the steganographic keys in the camera's memory would make the warden's task of detecting and extracting the embedded messages easier once he gains possession of the camera. Having a key pad to enter the key will likely arouse suspicion as to the existence of the keypad. Storing the keys on flash cards that can then be loaded and unloaded into the camera has to be carefully implemented. The warden can detect the keys on the flash card just as easily as if they were stored in the camera.

6 Conclusion

This paper introduced data embedding schemes for covert communications, digital watermarking, and information augmentation applications for digital images. It discussed the data embedding framework -- the assumptions implicitly and explicitly made on the communication channel, the implementation issues that are addressed during the design of a data embedding scheme, and the requirements that have to be balanced and formalized. For covert communication applications, undetectability is paramount. For source authentication applications, robustness to a variety of attacks is critical. For content authentication applications, one must be able to distinguish between malicious and non-malicious modifications. In a captioning application, capacity is maximized.

The paper also presented several examples of data embedding techniques for digital images. Spatial techniques are very sensitive to simple noise adding and filtering. Spectral techniques, based on spread spectrum techniques, are susceptible to desynchronization attacks. Robustness to geometric deformations is still an open issue and a solution must be found before watermarking techniques are successfully applied to image copyright protection. Limitations of these techniques are categorized into robustness attacks, presentation attacks, and interpretation attacks.

Several application scenarios are presented and the manner in which the data embedding framework is typically tailored and applied to them is proposed.

7 References

- [Adel99] A. Adelsbach, B. Pfitzmann, and A.-R. Sadeghi. "Proving Ownership of Digital Content." *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Ahum] A. Ahumada, Jr. and H. Peterson. "Luminance-Model-Based DCT Quantization for Color Image Compression." Preprint.
- [Ande96] R. J. Anderson, "Stretching the Limits of Steganography", *1st Information Hiding Workshop, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 39–48, 1996.
- [Ande98] R. J. Anderson and F. Petitcolas. "On the Limits of Steganography." *IEEE Journal on Selected Areas in Communications,* Vol. 16, No. 4, pp. 474-481, May 1998.
- [Algh99a] M. Alghoniemy and A. Tewfik, "Self-synchronizing Watermarking Technique" (invited paper), Proceedings of the Content Security and Data Hiding in Digital Media, NJIT Press, NJ, March 1999.
- [Algh99b] M. Alghoniemy and A. Tewfik, "Progressive Quantized Projection Watermarking Scheme," To be presented at the ACM'99 Multimedia conference, Orlando, FL, Nov. 1999.
- [Algh99c] M. Alghoniemy and A. Tewfik, "Synchronization Recovery in Image Watermarking," (invited paper) To be presented at the ACM'99 Multimedia conference, Orlando, FL, Nov. 1999.
- [Algh00] M. Alghoniemy and A. Tewfik, "Geometric Distortions Correction in image Watermarking," Submitted to the SPIE Security and Watermarking of Multimedia Contents, San Jose, CA, Jan. 2000.
- [Aura96] T. Aura. "Practical Invisibility in Digital Communication", In Ross Anderson ed., *Information Hiding*, pp. 7-21, Vol. **1174** of Lecture Notes in Computer Science, Springer, Berlin, 1996. (First International Workshop IH'96, Cambridge, UK, May/June 1996.
- [Bald99] A. Baldoza and M. Sieffert. "Methods of Detecting Tampering in Digital Images." Submitted to the Air Force Research Laboratory Technical Brief Program, 1999.
- [Bear98] D. Bearman and J. Trant, "Authenticity of Digital Resources: Towards a Statement of Requirements in the Research Process", *D-Lib Magazine*, June 1998.
- [Bend96] W. Bender, D. Gruhl, and N. Morimoto. "Techniques for Data Hiding." *IBM Systems Journal*, Vol. 35, No. 3-4, pp. 313-336, 1996.
- [Bola95] F. M. Boland, J. J. K. Ó Ruanaidh and C. Dautzenberg, "Watermarking digital images for copyright protection," *Proceedings of the International Conference on Image Processing and its Applications, Edinburgh*, Scotland, July 1995, pp. 321– 326.
- [Bone95] D. Boneh and J. Shaw, "Collusion-Secure Fingerprinting for Digital Data", *15th Annual International Cryptology Conference*, 27-31 August 1995, Santa Barbara, California, pp. 452-465.

- [Cach98] C. Cachin. "An Information-Theoretic Model for Steganography." *1st Information Hiding Workshop, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 39–48, 1996.
- [Cox95a] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. "Secure Spread Spectrum Watermarking for Multimedia." *NEC Research Institute Technical Report 95-10.*
- [Cox95b] I. Cox, S. Roy, and S. Hingorani. "Dynamic Histogram Warping of Image Pairs for Constant Image Brightness." In *IEEE International Conference on Image Processing*, 1995.
- [Cox96a] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. "A Secure, Robust Watermark for Multimedia." *Workshop on Information Hiding*, Cambridge, UK, May 1996.
- [Cox96b] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. "Secure Spread Spectrum Watermarking for Images, Audio, and Video." In *Proceedings of 1996 International Conference on Image Processing*, Lausanne, Switzerland, 16-19 September 1996, Vol. III, p. 243-246.
- [Cox97a] I. Cox and M. Miller. "A review of watermarking and the importance of perceptual modeling." *Proceedings of SPIE* 3016, *Human Vision and Electronic Imaging II*, San Jose, February 1997, pp. 92-99.
- [Cox97b] I. Cox and J. Linnartz. "Public watermarks and resistance to tampering." In *Proceedings of IEEE International Conference on Image Processing*, 1997. Paper appears only in CD versions of proceedings.
- [Cox98] I. Cox and J. Linnartz. "Some General Methods for Tampering with Watermarks." *IEEE Journal on Selected Areas in Communications* 16(4), 587-593 (May 1998).
- [Crav97] S. Craver, N. Memon, B. Yeo, and M. Yeung. "Can Invisible Watermarks Resolve Rightful Ownership?" In the *Proceedings of the IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases V*, San Jose, CA, USA, 13-14 February 1997, vol 3022, p. 310-321.
- [Crav99] S. Craver. "Zero-Knowledge Watermark Detection." *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Degu99] F. Deguillaume, G. Csurka, J. J. K. Ó Ruanaidh, and T. Pun, "Robust 3D DFT Video Watermarking", *Proceedings of SPIE Vol. 3657, Security and Watermarking of Multimedia Contents*, San Jose, California, January 1999.
- [Digi99] Digimarc Corporation. http://www.digimarc.com.
- [Egge99] J. Eggers and B. Girod. "Watermark Detection after Quantization Attacks." *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Etti98] Ettinger, J. M. "Steganalysis and Game Equilibria" Preprint.
- [Fran96] E. Franz, A. Jerichow, S. Moller, A. Pfitzmann, and I. Stierand. "Computer Based Steganography: How It Works and Why Therefore Any Restrictions on Cryptography Are Nonsense, at Best" In Ross Anderson ed., *Information Hiding*, pp. 7-21, Vol. **1174** of Lecture

Notes in Computer Science, Springer, Berlin, 1996. (First International Workshop IH'96, Cambridge, UK, May/June 1996.

- [Frid98a] J. Fridrich, A. Baldoza, and R. Simard. "Robust Digital Watermarking Based on Key-Dependent Basis Functions", *Proc. 2nd Information Hiding Workshop*, Portland OR, April 15-17, 1998.
- [Frid98b] J. Fridrich. "Methods for Detecting Changes in Digital Images", *Proc. of The 6th IEEE International Workshop on Intelligent Signal Processing and Communication Systems (ISPACS'98)*, Melbourne, Australia, 4-6 November 1998.
- [Frid98c] J. Fridrich. "Applications of Data Hiding in Digital Images", Tutorial for The *ISPACS'98* Conference in Melbourne, Australia, November 4-6, 1998.
- [Frid98d] J. Fridrich. "Image Watermarking for Tamper Detection", *Proc. ICIP '98*, Chicago, Oct 1998.
- [Frid98e] J. Fridrich. "Combining Low-frequency and Spread Spectrum Watermarking", *Proc. SPIE Int. Symposium on Optical Science, Engineering, and Instrumentation*, San Diego, July 19-24, 1998.
- [Frid99a] J. Fridrich. "Robust Bit Extraction From Images", submitted to *IEEE ICMCS'99* Conference, Florence, Italy, June 7-11, 1999.
- [Frid99b] J. Fridrich. "A New Steganographic Method for Palette-Based Images", submitted to The *IS&T PICS* Conference, Savannah, Georgia, April 25-28, 1999.
- [Frid99c] J. Fridrich. "Protection of Digital Images Using Self Embedding", with Miroslav Goljan, submitted to *The Symposium on Content Security and Data Hiding in Digital Media*, New Jersey Institute of Technology, March 16, 1999.
- [Frid99d] J. Fridrich and M. Goljan. "Comparing Robustness of Watermarking Techniques", *Proc. of SPIE vol. 3657 (Security and Watermarking of Multimedia Content)*, San Jose, Jan 25-27, 1999.
- [Frid99e] J. Fridrich and R. Du. "Secure Steganographic Methods for Palette Images", *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Hafe97] Mike Hafer. Final Report available at http://www.stanford.edu/~mah6023/final.htm.
- [Hart96] F. Hartung and B. Girod, "Digital Watermarking of Raw and Compressed Video", *Proc. European EOS/SPIE Symposium on Advanced Imaging and Network Technologies*, Berlin, Germany, October 1996.
- [Hart97a] F. Hartung and B. Girod, "Digital Watermarking of MPEG-2 Coded Video in the Bitstream Domain", *Proceedings ICASSP 97*, Vol. 4, pp. 2621-2624, Munich, Germany, April 1997.
- [Hart97b] F. Hartung and B. Girod, "Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video", in S. Fdida, M. Morganti (eds.), "Multimedia

Applications, Services, and Techniques – ECMAST '97", *Springer Lecture Notes in Computer Science*, Vol. 1242, pp. 423-436, Springer, Heidelberg, May 1997.

- [Hart97c] F. Hartung and B. Girod, "Fast Public Key Watermarking of Compressed Video", *Proceedings IEEE International Conference on Image Processing (ICIP 97)*, Santa Barbara, October 1997.
- [Hart98a] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video", *Signal Processing*, vol. 66, No. 3 (Special Issue on Watermarking), pp. 283-301, May 1998.
- [Hart98b] F. Hartung, P. Eisert, and B. Girod, "Digital Watermarking of MPEG-4 Facial Animation Parameters", *Computers & Graphics*, Vol. 22, No. 4 (Special issue on "Data Security in Image Communication and Network"), pp. 425-435, August 1998.
- [Hart99] F. Hartung, J. K. Su, and B. Girod, "Spread Spectrum Watermarking: Malicious Attacks and Counter-Attacks", *Proceedings of SPIE Vol. 3657, Security and Watermarking of Multimedia Contents*, San Jose, California, January 1999.
- [Herr98] A. Herrigel, J. J. K. Ó Ruanaidh, H. Petersen, S. Pereira and T. Pun, "Secure copyright protection techniques for digital images", In David Aucsmith ed., *Information Hiding*, pp. 169-190, Vol. **1525** of Lecture Notes in Computer Science, Springer, Berlin, 1998. (Second International Workshop IH'98, Portland, OR, USA, April 15-17, 1998.
- [Hons98] C. Honsinger and S. Daly. United States Patent 5,835,639 dated 10 Nov 1998.
- [Hsu96] C. Hsu and J. Wu. "Hidden Signatures in Images." In *Proceedings of IEEE International Conference on Image Processing*, Lausanne, Switzerland, 16-19 September 1996, vol. III, pp. 743-746.
- [Fran99] D. Frank and D. Brown. "White House shifts encryption strategy." Federal Computer Week, 20 Sep 99. Available at http://www.fcw.com/pubs/fcw/1999/0920/fcw-newsencrypt-09-20-99.html.
- [Geor99] M. George, J-Y. Chouinard, and N. D. Georganas, "Spread Spectrum Spatial and Spectral Watermarking for Images and Video", Proceedings of 1999 IEEE Canadian Workshop in Information Theory, Kingston, Ontario, June 1999.
- [Giro89] B. Girod, "The Information Theoretical Significance of Spatial and Temporal Masking in Video Signals", *Proc. of the SPIE Human Vision, Visual Processing, and Digital Display*, vol. 1077, pp. 178–187, 1989.
- [John98] N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen", *Computer*, Vol. 31, No. 2, February 1998, pp. 26-34.
- [John98b] N. Johnson and S. Jajodia. "Steganalysis of Images Created using Current Steganography Software," Proceedings of the Second Workshop on Information Hiding, Portland, OR, USA, 15-17 Apr 98. Lecture Notes in Computer Science, Vol. 1525, Springer-Verlag, 273.289. Available at http://www.jjtc.com/Steganalysis/.

- [John98c] N. Johnson and S. Jajodia. "Steganalysis: The Investigation of Hidden Information," IEEE Information Technology Conference, Syracuse, NY, USA, 1-3 Sep 98. Available at http://www.jjtc.com/Steganalysis/.
- [John99] N. Johnson. "An Introduction to Watermark Recovery from Images," Proceedings of SANS Intrusion Dection and Response (ID'99), San Diego, CA, 9-13 Feb 99. Available at http://www.jjtc.com/Steganalysis/.
- [John99b] N. Johnson, Z. Duric, and S. Jajodia. "Recovery of Watermarks from Distorted Images." *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Kalk] T. Kalker. "Watermark Estimation Through Detector Observations." Philips Research Eindhover, The Netherlands. Preprint.
- [Kalk97] T. Kalker, J. Linnartz, and M. van Kijk. "Watermark Estimation through Detector Analysis." In *Proceedings of the IEEE International Conference in Image Processing*, Santa Barbara, California, October 1997. Paper appears only in CD versions of proceedings.
- [Kerk1883] Kerkhoffs, "La Cryptographie Militaire", *Journals del Sciences Militaires*, 9th series, 1883.
- [Kahn96] D. Kahn, "The history of steganography", *1st Information Hiding Workshop, Lecture Notes in Computer Science*, R. Anderson, ed., vol. 1174, pp. 1–5, Springer-Verlag, 1996.
- [Kund97] D. Kundur and D. Hatzinakos. "A Robust Digital Image Watermarking Method using Wavelet-Based Fusion." In *Proceedings of the IEEE International Conference on Image Processing*, Santa Barbara, California, October 1997, vol. 1, 544-547.
- [Kund98a] D. Kundur and D. Hatzinakos. "Digital Watermarking Using Multiresolution Wavelet Decomposition." In the *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, Washington, vol. 5, pp. 2969-2972, May 1998.
- [Kund98b] D. Kundur and D. Hatzinakos. "Towards a Telltale Watermarking Technique for Tamper-Proofing", *Proceedings of the IEEE International Conference on Image Processing*, Chicago, October 1998.
- [Kura92] Kurak, C. and McHugh, J. "A Cautionary Note On Image Downgrading", *Proc. IEEE Eighth Annual Computer Security Applications Conference*, IEEE Press, Piscataway, NJ, 1992, pp. 153-159.
- [Kwan] Kwan, M. Gifshuffle. http://www.darkside.com.au/gifshuffle/
- [Lan99a] T-H Lan and A. Tewfik, "Fraud Detection and Self Embedding," To be presented at the ACM'99 Multimedia Conference, Orlando, FL, Nov. 1999.
- [Lan99b] T-H Lan and A. Tewfik, "High Capacity Data Embedding and Its Application in Fraud Detection," To be submitted to the IEEE Trans. on Image Processing.

- [Lin97] C-Y. Lin and S-F Chang, "A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation," *CU/CTR Technical Report 486-97-19*, December 1997.
- [Lin98a] C-Y. Lin and S-F Chang, "A Robust Image Authentication Method Surviving JPEG Lossy Compression", *SPIE International Conference on Storage and Retrieval of Image/Video Databases*, Vol. 3312. No. 37, Electronic Imaging 1998, San Jose, January 1998.
- [Lin98b] C-Y. Lin and S-F Chang, "Generating Robust Digital Signatures for Image/Video Authentication", *Multimedia and Security Workshop at ACM Multimedia 98*, Bristol, UK, September 1998.
- [Lin99] C-Y. Lin and S-F Chang, "Issues and Solutions for Authenticating MPEG Video", *SPIE International Conference on Storage and Retrieval of Image/Video Databases*, Vol. 3657. No. 06, Electronic Imaging 1999, San Jose, January 1999.
- [Linn] J. Linnartz and M. van Dijk. "Analysis of the Sensitivity Attack against Electronic Watermarks in Images." Philips Research Eindhoven, Eindhoven, The Netherlands. Preprint.
- [Mach] R. Machado, *EZ Stego*, [http://www.stego.com].
- [Marv98] L. Marvel, C. Boncelet, Jr., and C. Retter. "Reliable Blind Information Hiding for Images." In *Proceedings for the Second International Workshop in Information Hiding*, Portland, OR, 15-17 April 1998.
- [McCu99] McCullagh, D. "Clinton Relaxes Crypto Exports." Wired News. 16 Sep 99. Available at http://www.wired.com/news/news/email/explode-infobeat/politics/story/21786.html.
- [McDo99] A. McDonald and M. Kuhn. "StegFS: A Steganographic File System for Linux." *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Mint99] F. Mintzer and G. Braudaway, "If One Watermark is Good, Are More Better?" Preprint.
- [Mitt99] T. Mittelholzer. "An Information-Theoretic Approach to Steganography and Watermarking." *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Pere99] S. Pereira and T. Pun. "Fast Robust Template Matching for Affine Resistant Image Watermarks." *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Peti98] F. Petitcolas, R. Anderson, and M. Kuhn. "Attacks on Copyright Marking Systems." In *Proceedings for the Second International Workshop in Information Hiding*, Portland, OR, 15-17 April 1998.
- [Peti98b] F. Petitcolas. Stirmark. Available at http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/.

- [Peti99] F. Petitcolas, R. Anderson, and M. Kuhn. "Information Hiding -- A Survey." In *Proceedings for IEEE Special Issue on Protection of Multimedia Content*, 87(7):1062-1078, July 1999.
- [Pita96] I. Pitas. "A Method for Signature Casting on Digital Images." In *Proceedings of the International Conference on Image Processing*, Lausanne, Switzerland, September 1996, vol. III, p. 215-218.
- [Pita98] I. Pitas, "A Method for Watermark Casting on Digital Images", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 6, pp. 775-780, 1998.
- [Piva97] A. Piva, M. Barni, F. Bartolini, and V. Cappellini. "DCT-based Watermark Recovering without Resorting to the Uncorrupted Original Image." In *Proceedings of IEEE International Conference on Image Processing*, Santa Barbara, California, 26-29 October 1997, Vol. I, p. 520-523.
- [Piva98] A. Piva, M. Barni, F. Bartolini, and V. Cappellini. "Application-driven Requirements for Digital Watermarking Techniques", *European Multimedia, Microprocessor Systems and Electronic Commerce EMMSEC 98*, Bordeaux, France, 28-30 Sept 1998, in *Technologies for the Information Society: Developments and Opportunities*, J.-Y. Roger et al. (Eds.) IOS Press, 1998, pp. 513-520.
- [Podi98] C. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models", *IEEE Journal on Selected Areas I Communications*, Special Issue on Copyright and Privacy Protection, Vol. 16, No. 4, pp. 525-539, May 1998. Partially presented in *IS&T/SPIE Electronic Imaging: Human Vision and Electronic Imaging*, Vol. 3016, Feb 1997.
- [Qiao] L. Qiao and K. Nahrstedt. "Watermarking Schemes and Protocols for Protecting Rightful Ownership and Customer's Rights." Also available at http://cosimo.die.unifi.it/~piva/Watermarking/watermark_image.
- [Ramk98] M. Ramkumar and A. Akansu, "Information Theoretic Bounds for Data Hiding in Compressed Images," IEEE Second Workshop on Multimedia Signal Processing, Los Angeles, CA, USA, 7-9 Dec 1998.
- [Ramk98b] M. Ramkumar and A. Akansu, "A Robust Scheme for Oblivious Detection of Watermarks / Data Hiding in Still Images. SPIE Symposium on Voice, Video, and Data Communication, Boston, MA, Vol. 3528, pp 474-481, 2-5 Nov 1998.
- [Ramk99] M. Ramkumar and A. Akansu, "On the Choice of Transforms for Data Hiding in Compressed Video," IEEE International Conference in Audio, Speech, and Signal Processing, Phoenix, Arizona, March 1999.
- [Ruan96a] J. J. K. Ó Ruanaidh, W. Dowling, and F. Boland. "Watermarking Digital Images for Copyright Protection." *IEE Proceedings Vision, Image- and Signal Processing* 143(4), 250-256, August 1996. Available from http://cuiwww.unige.ch/~oruanaid/eva_pap.html.
- [Ruan96b] J. J. K. Ó Ruanaidh, W. Dowling, and F. Boland. "Phase Watermarking of Digital Images." In *Proceedings of IEEE International Conference on Image Processing*, Lausanne, Switzerland, 16-19 September 1996, vol. III, pp. 239-242.

- [Ruan97] J. J. K. Ó Ruanaidh, and T. Pun, "Rotation, scale and translation invariant digital image watermarking", *Proceedings of IEEE International Conference on Image Processing*, pp. 536-539, Santa Barbara, California, October 1997.
- [Ruan98] J. J. K. Ó Ruanaidh, and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking", *Signal Processing*, Vol. 66, No. 3, May 1998, pp. 373-383.
- [Ruan98b] J. J. K. Ó Ruanaidh and S. Pereira, "A secure robust digital image watermark", In *Electronic Imaging: Processing, Printing and Publishing in Colour*, SPIE Proceedings, Zürich, Switzerland, May 1998. (SPIE/IST/Europto Symposium on Advanced Imaging and Network Technologies).
- [Ruan99] J. J. K. Ó Ruanaidh and G. Csurka, "A Bayesian Approach to Spread Spectrum Watermark Detection and Secure Copyright Protection for Digital Image Libraries", To appear in *IEEE Conference on Computer Vision and Pattern Recognition*, Colorado, 23-25 Jun 1999.
- [Ruan99b] J. J. K. Ó Ruanaidh, "Watermarking methods", In *26th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'99)*, Los Angeles, CA, USA, 8-13 August 1999. Presented in the Panel ``Digital watermarking: what will it do for me? And what it won't!" (to appear)
- [Schn96] B. Schneier. "Applied Cryptography." Second Edition. John Wiley & Sons, Inc. 1996.
- [Serv98] S. Servetto, C. Podilchuk, and K. Ramchandran. "Capacity Issues in Digital Image Watermarking." Submitted to 1998 International Conference on Image Processing.
- [Shan48] C. E. Shannon, ``A mathematical theory of communication," *Bell System Technical Journal,* vol. 27, pp. 379-423 and 623-656, July and October, 1948.
- [Simm83] Simmons, G. "The Prisoner's Problem and the Subliminal Channel." Proceedings of CRYPTO '83, Plenum Press (1984), pp 51-67.
- [Smit96] J. R. Smith and B. O. Comiskey, "Modulation and information hiding in images", *Proceedings of the First International Workshop on Information Hiding*, R. Anderson ed. Lecture Notes in Computer Science, Vol. 1174, pp. 207-226, Springer Verlag 1996.
- [Su99] J. K. Su, F. Hartung, and B. Girod, "A Channel Model for a Watermark Attack", *Proceedings of SPIE Vol. 3657, Security and Watermarking of Multimedia Contents*, San Jose, California, January 1999.
- [Swan96a] M. Swanson, B. Zhu, and A. Tewfik. "Image Coding for Content-Based Retrieval." In *1996 SPIE Conference on Visual Communication and Image Processing.*
- [Swan96b] M. Swanson, B. Zhu, and A. Tewfik. "Transparent Robust Image Watermarking." In *Proceedings of IEEE International Conference on Image Processing*, Lausanne, Switzerland, 16-19 September 1996, vol. III, pp. 211-214.

- [Swan96c] M. Swanson, B. Zhu, and A. Tewfik. "Robust Data Hiding for Images." In *IEEE Digital Signal Processing Workshop*, Loen, Norway. September 1996, p. 37-40.
- [Tao97] B. Tao and B. Dickinson. "Adaptive Watermarking in the DCT Domain." In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Munich, Germany, 21-24 April 1997.
- [USCo] U.S. Copyright Office Web site is at http://lcweb.loc.gov/copyright/.
- [Voya96] G. Voyatzis and I. Pitas. "Chaotic Mixing of Digital Images and Applications to Watermarking." *European Conference on Multimedia Applications, services and Techniques* (ECMAST'96), Louvain-la-Neuve, Belgium, vol. 2, pp. 687-695, May 1996.
- [Wall91] G. Wallace. "The JPEG Still Picture Compression Standard." *Communications of the ACM*, vol. 34, no. 4, p. 30-44, 1991.
- [Walt95] S. Walton. "Image Authentication for a Slippery New Age", Dr. Dobb's Journal, April 1995. Available at http://www.ddj.com/articles/1995/9504/9504a/9504a.htm.
- [West99] A. Westfeld and A. Pfitzmann. "Attacks on Steganographic Systems: Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools -- and Some Lessons Learned." *Proceedings of Third Information Hiding Workshop*, Dresden, Germany, 29 Sep - 1 Oct, 1999.
- [Wolf96] R. Wolfgang and E. Delp. "A Watermark for Digital Images." In *Proceedings of IEEE International Conference on Image Processing*, Lausanne, Switzerland, 16-19 September 1996, vol. III, pp. 219-222.
- [Wolf97a] R. Wolfgang and E. Delp. "A Watermarking Technique for Digital Imagery: Further Studies." In the *Proceedings of the International Conference of Imaging Science, Systems, and Technology*, Las Vegas, Nevada, USA, 30 June – 3 July 1997, vol. 1, p. 279-287.
- [Wolf97b] R. Wolfgang and E. Delp. "Overview of image security techniques with applications in multimedia systems." *Proceedings of the SPIE Conference on Multimedia Networks: Security, Displays, Terminals, and Gatweys*, Vol. 3228, November 2-5, 1997, Dallas, Texas, pp. 297-308.
- [Wu98] M. Wu and B. Liu, "Watermarking for Image Authentication", *IEEE International Conference on Image Processing*, Chicago, October 1998.
- [Schy94] R. G. van Schyndel, A.Z. Tirkel, and C.F. Osborne, "A Digital Watermark," in *Proc. 1994 IEEE International Conference on Image Processing.*, vol. II, Austin, TX, 1994, pp. 86– 90.
- [Yeun97] M. Yeung and F. Mintzer, "An Invisible Watermarking Technique for Image Verification," *IEEE International Conference on Image Processing*, Santa Barbara, October 1997.

- [Zhu96] B. Zhu, M. Swanson, and A. Tewfik. "Transparent Robust Authentication and Distortion Measurement Techniques for Images." *7th IEEE Digital Signal Processing Workshop* , PP. 45-48, 1996.

8 Appendix A. Matlab code for Cox *et al* Data Embedding Encoder and Decoder.

The Matlab code for the Cox *et al* data embedding scheme was provided by Michael Hafer [Hafe97] and has been divided into several m-files:

Filename	Description
wm_example.m	Ties the other three supporting m- files into a coherent architecture
insertWaterMark.m	Inserts a watermark into a specified image
extractWaterMark.m	Extracts a watermark from a specified image
compareWaterMarks .m	Performs the similarity function needed to compare watermarks

8.1 wm_example.m

```
% Filename: wm_example.m
% Author: Mike Hafer, 12/98
% The following code shows an example of the watermarking and
% extraction process

im_org = pgmRead('einstein.pgm');

% Create a random watermark of length 1000
waterMark = randn(1000, 1);
wmStrength = 0.1;

% Create watermarked version of Einstein image
[im_WM, wInd] = insertWaterMark(im_org, waterMark, wmStrength);

% Generate recovered (corrupted) version of watermarked Einstein image
im_recovered = im_WM;
im_recovered(90, 20) = 500;

% Extract watermark from recovered image
wm_ext = extractWaterMark(im_org, im_recovered, wInd, wmStrength);

% Get watermark detector reading for the recovered image
similarity = compareWaterMarks(waterMark, wm_ext);
```

8.2 insertWatermark.m

```
% Filename: insertWaterMark.m
% Author: Mike Hafer, 12/98
% [im_WM, WMind] = insertWaterMark(im, WaterMark, strengthWM)
%
% Inserts a watermark into an image to produce a water marked
% image.  This is a modified version of insertWaterMarkOrg.
% This version excludes the pure vertical and horizontal DCT
% elements from the perceptual mask.
%
% im: Image to be water marked, D
%
% WaterMark: Water mark to add to image, X
%
% strengthWM: Water mark strength
%
% im_WM: Water marked image, D'
%
% WMind: Water mark indices for V'
%

function [im_WM, WMind] = insertWaterMark(im, WaterMark, strengthWM)

lengthWM = length(WaterMark);

% Determine the perceptually significant elements, V, of the
% image's DCT.  This is done by finding the indices corresponding
% to the largest absolute DCT coefficient values excluding
% the pure vertical and horizontal DCT coefficients. The number of
% indices equals "lengthWM"
im_DCT = dct2(im);
im_DCT_mod = im_DCT;
im_DCT_mod(:,1) = zeros(size(im_DCT, 1), 1);  % Exclude pure vertical coefficients of DCT
im_DCT_mod(1,:) = zeros(1, size(im_DCT, 2));  % Exclude pure horizontal coef of DCT
im_DCT_vec = im_DCT_mod(:);
sorted_DCT = flipud(sort(abs(im_DCT_vec)));
lower_bound = sorted_DCT(lengthWM);
ind1 = find(abs(im_DCT_vec) > lower_bound);
ind2 = find(abs(im_DCT_vec) == lower_bound);
WMind = [ind1; ind2(1:(lengthWM - length(ind1)))]; % DCT indices for V & V'

% Generate water marked DCT element, V':
```

```
%   V' = V*(1 + strengthWM*X)
im_DCT_WM = im_DCT;
V = im_DCT(WMind);
V_prime = V.*(1 + strengthWM*WaterMark);

% Generate DCT for water marked image.
im_DCT_WM(WMind) = V_prime;

% Generate the water marked image, D'
im_WM = idct2(im_DCT_WM);
```

8.3 extractWaterMark.m

```
% Filename: extractWaterMark.m
% Author: Mike Hafer, 12/98
% [WaterMark] = extractWaterMark(imOrg, imWM, WMind, strengthWM);
%
% Extracts the water mark from an image.  This version
% rescales the DCT perceptual elements, V*, from the recovered image
% to match the norm of orginal image's DCT perceptual
% elements, V.
%
% imOrg: Original unwatermarked image, D
%
% imWM: Recovered watermarked image, D*
%
% WMind: Watermark DCT indices for V and V*
%
% strengthWM: Watermark strength
%
% WaterMark: Extracted watermark, X*

function [WaterMark] = extractWaterMark(imOrg, imWM, WMind, strengthWM);

% Extract the DCT perceptual elements, V, from the original image, D
DCT_org = dct2(imOrg);
V_org = DCT_org(WMind);

% Extract the DCT perceptual elements, V*, from the recovered image, D*
DCT_WM = dct2(imWM);
V_WM = DCT_WM(WMind);

% Rescale the DCT perceptual elements, V*, from the recovered image
V_WM = V_WM * norm(V_org)/norm(V_WM);
```

```
% Extract the watermark using the equation:
%   X* = (V* - V)/(V*strengthWM)
WaterMark = (V_WM - V_org)./(V_org.*strengthWM);
```

8.4 compareWaterMarks.m

```
% Filename: compareWaterMarks.m
% Author: Mike Hafer, 12/98
% [similarity] = compareWaterMarks(WM_org, WM_extracted);
%
% Evaluates the similarity of watermarks
%
% WM_org: Original watermark, X
%
% WM_extracted: Extracted watermark, X*
%
% similarity: Similarity reading;
%             Value of 1.0 indicates an exact match
%

function [similarity] = compareWaterMarks(WM_org, WM_extracted);

similarity = WM_extracted'*WM_org/norm(WM_extracted)/norm(WM_org);
```

9 Appendix B. Matlab code for [Ruan98] Data Embedding Encoder and Decoder.

The Matlab code for the [Ruan98] data embedding scheme was provided by Jiri Fridrich and has been divided into several m-files:

Filename	Description
PivaTest.m	Ties the other three supporting m- files into a coherent architecture
PivaWmk.m	Inserts a watermark into a specified image
PivaDetect.m	Extracts a watermark from a specified image
compare.m	Performs a [Giro89] comparison between two specified images in order to compute the percentage of pixels with visible differences
genMsg.m	Generates a key-based message
localstd.m	Calculates the local standard deviation for a matrix.
getindexy.m	Returns vector of indices of the first D diagonally chosen matrix bins following D0 omitted bins

9.1 PivaTest.m

```
function PivaTest (image, output_image, L, M, alpha, seed );
% Watermark Image
[Xw, betabar, alphabar]=PivaWmk(image, output_image, L, M, alpha, seed);

% Compare Original with Watermarked Image
[X,Map]=bmpread(image);
[Xw,Map]=bmpread(output_image);
[pixels,percent_bad]=compare(X,Xw,0.003);
fprintf ('--->Percent Change Between Image and Marked Image = %f\n', percent_bad);

% Detect Watermark
[z,Sz,Tro]=PivaDetect(output_image, L, M, alphabar, seed);
```

9.2 PivaWmk.m

```
function [Xw,betabar,alphabar]=PivaWmk(image,output_image,L,M,alpha,seed);

% Watermarking method by Piva from A. Piva, M. Barni, F. Bartolini,
% V. Cappellini, "DCT-based watermark recovering without resorting
% to the uncorrupted original image", preprint, 1997.
%
% Adjustment for visibility is done using simple localized std
% This function uses localstd.m for calculating localized std
% and getindexy.m for selecting DCT coeffs in a zig-zag manner
%
% Inputs:
% image   Input image to be watermarked (in BMP format)
% L       First L DCT coefficients will be skipped
% M       The next M DCT coefficients will be modified
% alpha   Watermark strength/visibility
% seed    Seed for a PRNG
%
% Output: Watermarked image Xw
% Typical usage: Xw=piva1new('pep512.bmp',25000,16000,2,seed);
%
% Piva recommends using L=25000, M=16000, and alphabar=0.2, alpha=alphabar/betabar
%  The visibility of the watermark stays below an acceptable level
%  (according to spatial masking model of Girod). To test that, type:
%  compare('pep512.bmp','water.bmp',0.003)   right after running piva1.m.

  [X,Map]=bmpread(image);
  [N1,N2]=size(X);
  original=X;
  rs=getindexy(N1,N2,L,M);     % Diagonal indices of the first M
                               % except the first L DCT bins
  DX=dct2(X);                  % Calculate the 2D DCT transform of X
  DXw=DX;

% Watermarking-beginning

  randn('state',seed);
  eta=randn(M,1);
  DXw(rs)=DX(rs)+alpha*(eta.*abs(DX(rs)));

  Xw=idct2(DXw);
  Xw(find(Xw>256))=256;                % Take care of boundary effects
  Xw(find(Xw<1))=1;
```

```
% Adjustment for visibility

  K=9;                         % Kernel size for calculating local std
  s=localstd(X,K);             % Matrix of local stds
  beta=s/max(max(s));          % Correction factor beta
  Xw=beta.*Xw+(1-beta).*X;     % Adjustment for visibility (beta~1 emphasizes the
                               % watermarked image, beta~0 emphasizes the original)
% Show the original image
  figure(1);imshow(original,Map);
  title('Original image');

% Show the watermarked image
  figure(2);imshow(Xw,Map);
  title('Watermarked image');
  bmpwrite(Xw,Map,output_image);      % Store watermarked image to disk as 'water.bmp'

% Information about the energy of the watermark
  df=Xw-original;
  powerofwm=mean2(df.^2);
  fprintf('  The average watermark power is:%f\n', powerofwm);
  fprintf('  %d < Watermark < %d\n', floor(min(min(df))), floor(max(max(df))));

% Control output (see the paper by Piva)
betabar=mean2(beta);
alphabar=alpha*betabar;        % this alphabar become constant in detections!
```

9.3 PivaDetect.m

```
function [z,Sz,Tro]=PivaDetect(image,L,M,alphabar,seed);

% Watermarking method by Piva from A. Piva, M. Barni, F. Bartolini,
% V. Cappellini, "DCT-based watermark recovering without resorting
% to the uncorrupted original image", preprint, 1997.
%
% image     Watermarked, attacked image
% L         First L DCT coefficients will be skipped
% M         The next M DCT coefficients will be modified
% alphabar  Watermark strength - for calculation Sz threshold
% seed      Seed for a PRNG
% Typical usage: [z Sz Tro]=det1new(image,25000,16000,0.2,seed);
%  for image size other than 512x512 L=25000*N1/512*N2/512 M=16000*N1/512*N2/512

  [X,Map]=bmpread(image);
  [N1,N2]=size(X);
  original=X;
  DX=dct2(X);                  % Calculate the 2D DCT transform of X

% Detection-beginning

  rs=getindexy(N1,N2,L,M);     % Diagonal indices of the first M
                               % except the first L DCT bins

  randn('state',seed);
  eta=randn(M,1);

  Sz=sum(abs(DX(rs)));         % Image dependent threshold
  Sz=Sz*alphabar/(3*M);
  z=sum(DX(rs).*eta);
  z=z/M;
  stdt=std(DX(rs));            % An alternative threshold follows
  Tro=3.3*sqrt(2/M*stdt*stdt);% from "Threshold Selection ..." Piva's paper
```

9.4 compare.m

```
function [pixels,percent_bad]=compare(image1,image2,Th);

% This function is used to find out if differences between two images are visible.
% The model used is the spatial masking model described by Girod. The inputs are:
% two images in BMP format, and a value of the threshold Th for the saturation
% signal. Based on computer experiments using threshold.m this value should be about 0.003.
% The function decides whether or not the two images are perceptually identical.
% The areas containing visible differences are shown in black.
%
% A simplifying assumption of a constant, homogenous JND in the saturation signal is used
% to invert Girod's spatial masking model. For more details, consult Girod's paper.
%
% X1 .... image1
% X2 .... image2
% Th .... threshold
% Typical usage: compare('len256.bmp','water.bmp',0.003);

%[X1,Map1]=bmpread(image1);
X1=image1;
[M1,N1]=size(X1);
%[X2,Map2]=bmpread(image2);
X2=image2;
[M2,N2]=size(X2);

if N1~=N2 | M1~=M2
```

```
  disp('** Warning: **');
  disp('Images have different dimensions.');
  disp('Truncating dimensions ...');
end

N=min(N1,N2);
M=min(M1,M2);

% Viewing characteristics

  v=4;                                 % Viewing distance of v image diagonals
  Alpha=(180/pi)*2*atan(1/(2*v));      % Angle subtended by the image diagonal (in degrees)
  Ppd=(sqrt(M^2+N^2)/Alpha)/60;               % Ppd = pixels per arc minute

% Parameters of the spatial masking model (according to Girod's paper)

  Lmon=0.00035;Smon=15;G=2.2; % Monitor characteristics provided it complies with CCIR
  Ksat=8;                     % Saturation coefficient
  Lad=7;                      % Level of ambient illumination
  Sinh=8;                     % Spatial inhibition spread in arc minutes
  Savg=13;                    % Saturation averaging spread in arc minutes

% Spatial masking

  L=Lmon*power(X1+Smon,G);             % Conversion from grayscale to screen luminance
  w1=G*Lmon*power(X1+Smon,G-1);               % Factor w1 for the linearized model
  dL=w1.*(X2-X1);
  h=fspecial('gaussian',[10 10],Sinh*Ppd);    % Gaussian filter 10x10 with sigma=Sinh [arcmin]
  Linh=Lad+conv2(L,h,'same');                 % Inhibited luminance Linh (see Girod's paper)
  w2=ones(M,N)./(Linh+Ksat*max(zeros(M,N),L-Linh)); % Factor w2 for the linearized model
  h=fspecial('gaussian',[10 10],Savg*Ppd);    % Averaging Gaussian kernel with sigma=Savg [arcmin]
  dcsat=dL.*w2;                        % Increment in fovea saturation
  dcsat2=dcsat.*dcsat;        % Square of the saturation increment
  dS=filter2(h,dcsat2);                % Averaging the saturation increment using h

% Determining the pixels with visible changes

  bad_pixels=find(dS>Th);
  D=ones(M,N);
  D(bad_pixels)=0;

% Show image1 and 2

  subplot(1,3,1);imshow(X1,Map1);
  title('Image1');
  subplot(1,3,2);imshow(X2,Map2);
  title('Image2');

% Show the bad pixels in black, the rest in white

  subplot(1,3,3);
  imshow(D);
  title('Black pixels = visible changes');
  bad=length(bad_pixels);
  fprintf('  %d pixels (%f percent) have visible changes\n', bad, bad*100/(M*N));

  percent_bad = bad*100/(M*N);
  pixels=length(bad_pixels);
```

9.5 genMsg.m

```
function message=genMsg(numberOfSymbols);

msg='X';
rand ('state', 13);
for count=1:(numberOfSymbols+1)
   if (rem(rand(1),2) > 0.5)
      msg = strcat (msg, '1');
   else
      msg = strcat (msg, '0');
   end
end

message = msg(2:length(msg)-1);
```

9.6 localstd.m

```
function s=localstd(X,K);

% This function calculates the local standard deviation for matrix X.
% The standard deviation (std) is evaluated for a square region KxK
% pixels surrounding each pixel. At the boundary, the matrix is NOT
% padded. Instead, the std is calculated from available pixels only.
% If K is not an odd integer, it is floored to the closest odd integer.
%
% Input:  MxN matrix X
%         K   size of the square region for calculating std
% Output: s   local std calculated for KxK regions
% Typical use: s=localstd(X,3);
```

```
[M N]=size(X);

if mod(K,2)~=1                % K must be an odd number
  K=K+1;
end

kern=ones(K,K);                       % Kernel for calculating std
x1=conv2(X,kern,'same');      % Local sums
x2=conv2(X.*X,kern,'same');   % Local quadratic sums
R=conv2(ones(M,N),kern,'same');       % Number of matrix elements in each square region
s=sqrt(x2./R-(x1./R).^2);     % Local standard deviation

% An old approach using blockprocessing - it cannot handle matrices larger than 500x500
% and incorrectly calculates std at the boundaries
% W=zeros(M+2,N+2);
% W(2:M+1,2:N+1)=X;
% z=zeros(M,N);
% z(:)=std(im2col(W,size(kern),'sliding'),1);
```

9.7 getindexy.m

```
function [ind,DIAG]=getindexy(M,N,D0,D);

% This function returns vector of indices of the first D diagonally chosen
% matrix bins following D0 omitted bins
% MxN matrix
% Scanning order for 4x5:
% 1  3  6 10 14 18
% 2  5  9 13 17 21
% 4  8 12 16 20 23
% 7 11 15 19 22 24
% Typical usage: ind=getindexy(512,512,25000,16000);
% The second output, DIAG is usually not needed

DIAG=(1:M)'*ones(1,N) + ones(M,1)*(1:N);     % row+col matrix
% 2 3 4 5 6 7
% 3 4 5 6 7 8
% 4 5 6 7 8 9
% 5 6 7 8 9 10
DIAG=(DIAG-2).*(DIAG-1)/2 + ones(M,1)*(1:N);
Qo=((1:M)'*ones(1,N))+(ones(M,1)*(1:N));

if M==N
```

```
 P=Qo-(M+1);          % correction for the lower down corner part of the matrix
 P(P<0)=0;
 Z=P.*P;
 DIAG=DIAG-Z;
else
 mi=min(M,N);         % correction for the center paralelogram part of the matrix
 P=Qo-(mi+1);
 P(P<0)=0;
 Z=P.*(P+1)/2;
 DIAG=DIAG-Z;

 ma=max(M,N);         % correction for the lower down corner part of the matrix
 P=Qo-(ma+1);
 P(P<0)=0;
 Z=P.*(P+1)/2;
 DIAG=DIAG-Z;

 P=Qo-(N+1);          % correction for "N+" part of the matrix
 P(P<0)=0;
 DIAG=DIAG+P;
end

ind=find(DIAG<=D+D0 & DIAG>D0);
```

☆U.S. GOVERNMENT PRINTING OFFICE: 2000-510-079-81251

MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.